The development and implementation of variable tip-sample separation scanning tunnelling spectroscopy to enhance spectroscopic measurements.

Franks, James R

The Development and Implementation of Variable Tip-Sample Separation Scanning Tunnelling Spectroscopy to Enhance Spectroscopic Measurements.

James R. Franks, B_{ENG}.

A thesis submitted for the degree of
Doctor of Philosophy, in the School
Engineering.

University of Wales, Swansea.

September 2003.

TABLE OF CONTENTS

DECLARATION

The work contained in this thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed _____
(candidate)

The work contained in this thesis, other than where stated within, is the result of the candidates own investigation, performed under the guidance of the director of studies named below.

Signed _____
(candidate)

Signed _____
(Director of Studies)

ACCESS TO CONTENTS

PRIFYSGOL CYMRU UNIVERSITY OF WALES

Author:

JAMES ROBERT FRANKS

Title of Thesis:

The Development and Implementation of Variable Tip-Sample Separation Scanning Tunnelling Spectroscopy to Enhance Spectroscopic Measurements.

I hereby give my permission for my thesis, if accepted, to be made available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed _____

Date 27th feb 2004

ACKNOWLEGMENTS

I would like to express my warmest gratitude to the director of studies, Professor S. Wilks, for his direction and support throughout the course of my studies.

I am indebted to my colleagues within the Multi Disciplinary Nano Technology centre in Swansea for there advice and support. In particular I recognise the effort of Dr Thierry Maffeis and Martin Brown for their help developing data processing software, and Dr Vincent Teng for his many and wide ranging contributions.

I would like to recognise Dr T Davies and Andrew Jardine for their many insightful comments and ideas, and Mark Pritchard, Tony Green and Jeff Mort for their technical support.

Finally, my deepest gratitude is reserved for my wife Sarah and my parents Robert and Janet Franks and my brother Andrew Franks and my Saviour Jesus Christ, who have not faltered in their love and encouragement.

ABSTRACT

Scanning Tunnelling Microscopy (STM) and Spectroscopy (STS) are valuable surface science techniques in the characterisation of a wide variety of surfaces on the atomic scale. When STM and STS are performed concurrently, both the topological and electronic properties of a surface can be investigated.

Conventionally STS is performed with the tip-sample separation fixed, which leads to a limited dynamic range in the measurement. This limitation can be overcome by varying the tip-sample separation during STS. Varying the tip-sample separation improves the dynamic range by exploiting the exponential relationship between tunnel current and tip-sample separation.

To accomplish variable tip-sample separation STS with Omicron Micro STM and STM/SEM-HC Scanning Tunnelling Microscope systems, external instrumentation was designed and fabricated. This instrumentation operated in conjunction with the Omicron microscopes to create the tip-sample contour and measure conductivity during variable tip-sample separation STS.

The instrumentation performed as expected, although problems were encountered making conductivity measurements using a Lock-in amplifier. Conductivity spectra obtained using a Lock-in amplifier contained perturbations related to the tip-sample capacitance. Future work is required to compensate for the effects of the tip-sample capacitance in conductivity measurements based around a Lock-in amplifier.

To comparatively evaluate variable and fixed tip-sample separation techniques, STS was performed consecutively on Si(111) 7x7 and GaAs(110) surfaces. For both surfaces an improvement in the definition of spectral features is observed. In the case of GaAs(110), surface states indistinguishable from noise in fixed tip-sample separation STS are clearly observed utilising variable tip-sample separation.

Chapter 1

Introduction

Introduction

Throughout the past few decades considerable advancements in electronic technologies have been made, to the extent that modern society increasingly relies on these technologies. Electronic products pervade modern life, finding applications in areas as diverse as medicine and children's toys. The proliferation of electronics based products is intimately linked with improvements in the semiconductor devices they use, and the manufacturing technology used to fabricate them. Intrinsic to improvements in semiconductor devices has been a greater understanding of semiconductors on the atomic scale.

In real semiconductor devices the interface between the device and external connections plays an important role in determining the ultimate performance of the device. It is necessary to study the surface properties of semiconductors, because they differ significantly from the bulk properties and are currently not well understood. The difference between bulk and surface properties originates from the different bonding environments experienced by bulk and surface atoms. Often the differing properties cause the introduction of electronic states, termed 'surface states' [1] within the bulk bandgap of the device, adversely affecting the performance of the device. As an example, the introduction of surface states within an AlGaAs-GaAs III-V semiconductor laser device, such as is used in optical storage, causes non-radiative recombination to occur, which results in local heating and ultimately destruction of the device [2].

The trend within the microelectronics industry to produce faster and smaller devices necessitates an even grater understanding of the role that surfaces play in device performance; to this end surface analysis techniques are continually evolving. One technique which has been used with great success to study semiconductors on the atomic scale is Scanning Tunnelling Microscopy (STM).

Since the first atomic resolution images of Si(111) 7x7 were presented by Binning and Rohrer [3] in 1982, STM has become one of the most powerful surface analysis techniques available to characterise structural and electronic properties of surfaces on the atomic scale. Many developments have been made in the two decades following

the work of Binning and Rohrer, as STM has evolved and spawned new Scanning Probe Microscopy (SPM) techniques. Today SPM techniques are widely used to study a diverse variety of materials, ranging form DNA and biological material to superconductors; in environments ranging from liquid to ultra high vacuum.

A technique that often accompanies and compliments STM is Scanning Tunnelling Spectroscopy (STS). Inherent in STM is the quantum mechanical tunnelling [4] process between two electrodes, usually an atomically sharp tip and sample. By varying the potential difference between the tip and sample, the states that take part in the tunnelling process can be investigated. When STS is performed concurrently with STM, electronic properties of the surface can be correlated with structural features.

To make STM measurements the aforementioned tip, usually chemically etched from Tungsten, is brought within a few angstroms of the sample material surface and then scanned across the sample surface. To achieve this, the tip is mounted on a scanner manufactured from a piezoelectric material [5]. The scanner is constructed in such a way to allow the tip position to be controlled in three dimensions. When a voltage is applied to the scanner it moves the tip with atomic precision. In order to position the tip within a few angstroms of the sample surface the current due to quantum mechanical tunnelling between the tip and sample is monitored as the tip is moved toward the sample surface. When the magnitude of the tunnelling current attains a preset value the approach is stropped. During tip approach and STM imaging a potential difference between the tip and sample is present. This potential difference is created by applying a bias voltage to either the sample or the tip. The potential difference ensures the Fermi levels of the tip and sample are at different energies, resulting in a net tunnelling current.

In the most commonly used STM imaging mode, the tunnelling current is constantly monitored as the tip is scanned across the sample surface. The separation between the tip and sample surface is adjusted to maintain a constant preset tunnelling current. The surface topography is inferred from the vertical movement of the tip required to maintain this preset tunnelling current. To create a topographic image of an area of the sample surface a number of parallel scans are performed, sequentially building an STM image.

Conventionally, to perform STS measurements whilst imaging using STM the scan is momentarily interrupted and the tip positioned at a site of interest. Adjustment of the tip-sample separation to maintain a constant tunnel current is disabled, thus the tip-sample separation remains fixed throughout the measurement. The bias applied to either the tip or sample is ramped in discrete steps and tunnel current measurements performed for every step [6]. The result of this procedure is a current-voltage spectra, which may be mathematically processed to access the effective density of electronic states available at a particular energy.

The tunnelling current spans several orders of magnitude when STS measurements are performed in a conventional manner on semiconducting samples. This large dynamic range of tunnel current is caused by the presence of a bandgap in the semiconductor. The finite signal to noise ratio of the tunnel current measuring instrumentation limits resolution within the bandgap. Surface state features that occur within the bandgap are effectively veiled by noise due to limitations in measurement resolution.

To overcome resolution limitations in conventional STS the tip-sample separation can be varied during STS measurements [7]. This technique exploits the exponential nature of the relationship between tunnel current and tip-sample separation [8]. By reducing the tip-sample separation the tunnel current is effectively amplified to a conveniently measurable value. Data acquired using the variable tip-sample separation technique requires more complex mathematical treatment [9] than data acquired with conventional STS. This is because the measured current includes contributions from varying the tip-sample separation.

To implement the variable tip-sample separation STS technique using standard Omicron STM systems necessitated the design and manufacture of additional instrumentation. This instrumentation is responsible for generating a suitable tip-sample separation contour and measuring conductivity during STS measurements. The detailed design and test of such instrumentation forms the basis of this PhD study.

Results are presented for fixed and variable tip-sample separation STS measurements performed on Si(111) 7x7 and GaAs(110) using the designed instrumentation. In both cases measurements made using variable tip-sample separation show a marked improvement in definition of spectral features. In the case of GaAs(110) additional features are observed within the bandgap region, which were veiled by noise in fixed tip-sample separation STS measurements.

The ability displayed by variable tip-sample separation STS to resolve features within the bandgap of semiconductor materials provides device designers with a technique that aids further understanding of the contribution these features make to device performance, and therefore improve the design of devices.

1.1 Thesis outline.

Chapter 2 – A presentation of the basic principles and theory on which STM and STS are founded is made. The quantum mechanical tunnelling phenomena exploited by the scanning tunnelling microscope is introduced, followed by the basic operating principles of a practical microscope. The theory underpinning STM is introduced and examples of STM on Si(111) 7x7 and GaAs (110) given. STS theory is introduced and fixed and variable tip-sample separation methods discussed. An example of the application of variable tip-sample separation STS to Sb covered GaAs(110), as previously studied by Feenstra *et al* [10] is given.

Chapter 3 – The Omicron Micro-STM and STM/SEM-HC microscopes utilised in experimentation are described, along with the SCALA control electronics which is common to both. An overview of the SCALA Pro software utilised by both microscopes is undertaken. A comprehensive description of the facilities provided by the SCALA electronics and SCALA Pro software to facilitate the interface of external instrumentation is given. Finally, the electrochemical etching method used to prepare tips for use by both microscopes is described.

Chapter 4 - The design of instrumentation to implement variable tip-sample separation STS in conjunction with the Omicron SCALA electronics is presented. A general analysis of the requirements of variable tip-sample separation STS from a

hardware perspective is given and a block level solution introduced. Detailed descriptions of the design of each functional unit within the block level solution are presented.

Chapter 5 – The development and function of software created for the instrumentation used to implement variable tip-sample separation STS is presented. Software tools and programming languages used during the development process are introduced and briefly described.

Chapter 6 – The instrumentation and software developed previously is evaluated by comparing measurements made at various circuit locations with theoretical expectations. Problems encountered with conductivity measurements are discussed.

Chapter 7 – Results are presented for STM, and fixed and variable tip-sample separation STS studies of p-type Si(111) 7x7 and p-type GaAs(110). In each case sample preparation techniques are addressed firstly followed by discussion of STM results and finally discussion of fixed and variable tip-sample separation STS results

Chapter 8 - Conclusions are drawn from the work completed and recommendations for future work discussed.

1.2 References.

[1] J.E. Inglesfield, Rep. Prog. Phys, **45**, 223 (1982).

[2] O. Ueda, J. Electrochem. Soc.: Reviews and News, **135**(1), 11C (1988).

[3] G. Binning, H. Rohrer, Ch. Gerber, and E. Weibel, Phys. Rev. Lett, **49**, 57 (1982).

[4] S.O. Kasap, 'Principles of Electrical Engineering Materials and Devices', Chapter 3, **183 – 188**, McGraw Hill (1997).

[5] S. I. Park, R.C. Berrett, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 2, **53-56**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[6] J.A. Stroscio, R.M. Feenstra, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 4, **97 - 99**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[7] J.A. Stroscio, R.M. Feenstra, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 4, **134 - 138**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[8] D.A. Bonnell, B.D. Huey, 'Scanning Probe Microscopy and Spectroscopy', Chapter 2, **8 - 10**, Edit D. Bonnell, Second edition, Wiley-VCH (2001).

[9] C.K. Shih, R.M. Feenstra, and P. Martensson, J. Vac. Sci. Technol.,A **8**, 3379 (1990).

[10] R.M. Feenstra and P. Martenesson, Phys. Rev. Lett. **61**, 447 (1988).

Chapter 2

STM and STS Theory

Introduction

Herein a brief introduction to Scanning Tunnelling Microscopy (STM) and Spectroscopy (STS) is presented, along with examples of STM application to the well understood Si(111) 7x7 and GaAs(110) surfaces. Additionally, examples of the application of STS to metal adsorbates on the surface of GaAs(110) is presented.

The basic operation of STM and the underlying quantum tunnelling phenomena exploited by STM is discussed in section 2.1. In section 2.2 the theoretical interpretation of STM is discussed along with constant current and constant height operating modes. In section 2.3 the theoretical interpretation of STS is presented along with details of fixed tip-sample separation and variable tip-sample separation acquisition methods.

2.1 Basic principles.

2.1.1 Quantum tunnelling.

The quantum mechanical phenomenon of 'tunnelling' is exploited as the foundation principle on which STM is based. When two conducting electrodes separated by a vacuum barrier are brought into close proximity, an electron incident on the barrier has a finite probability of transmission across the barrier due to its wave like nature [1]. This results in current flow between the conducting electrodes due to tunnelling, which in the case of STM are the tip and sample. The current decays exponentially with barrier width [2] as defined by equation 2.1.

$$I \propto e^{-2kZ} \tag{2.1}$$

Here the tunnel current I, depends exponentially on the wave vector k, which is a function of energy and the tip-sample separation Z.

Figure 2.1.1 illustrates schematically the tunnelling processes between a metallic tip and a semiconducting sample under differing bias conditions. The metallic tip is considered to possess a constant density of states at the Fermi level, whereas the semiconductor possesses surface states. Figure 2.1.1 (a) shows the tip and sample in

thermal equilibrium and separated by a vacuum barrier of appropriate width to promote tunnelling. This situation results in no net current flow since the Fermi levels of the tip (E_{FT}) and sample (E_{FS}) are at the same energy. In figure 2.1.1 (b) a positive bias is applied to the sample, which results in a net current flow arising from electrons occupying states in the tip tunnelling to unoccupied states in the sample. In figure 2.1.1 (c) a negative bias is applied to the sample, which results in a net current flow from occupied states in the sample to unoccupied states in the tip [3].

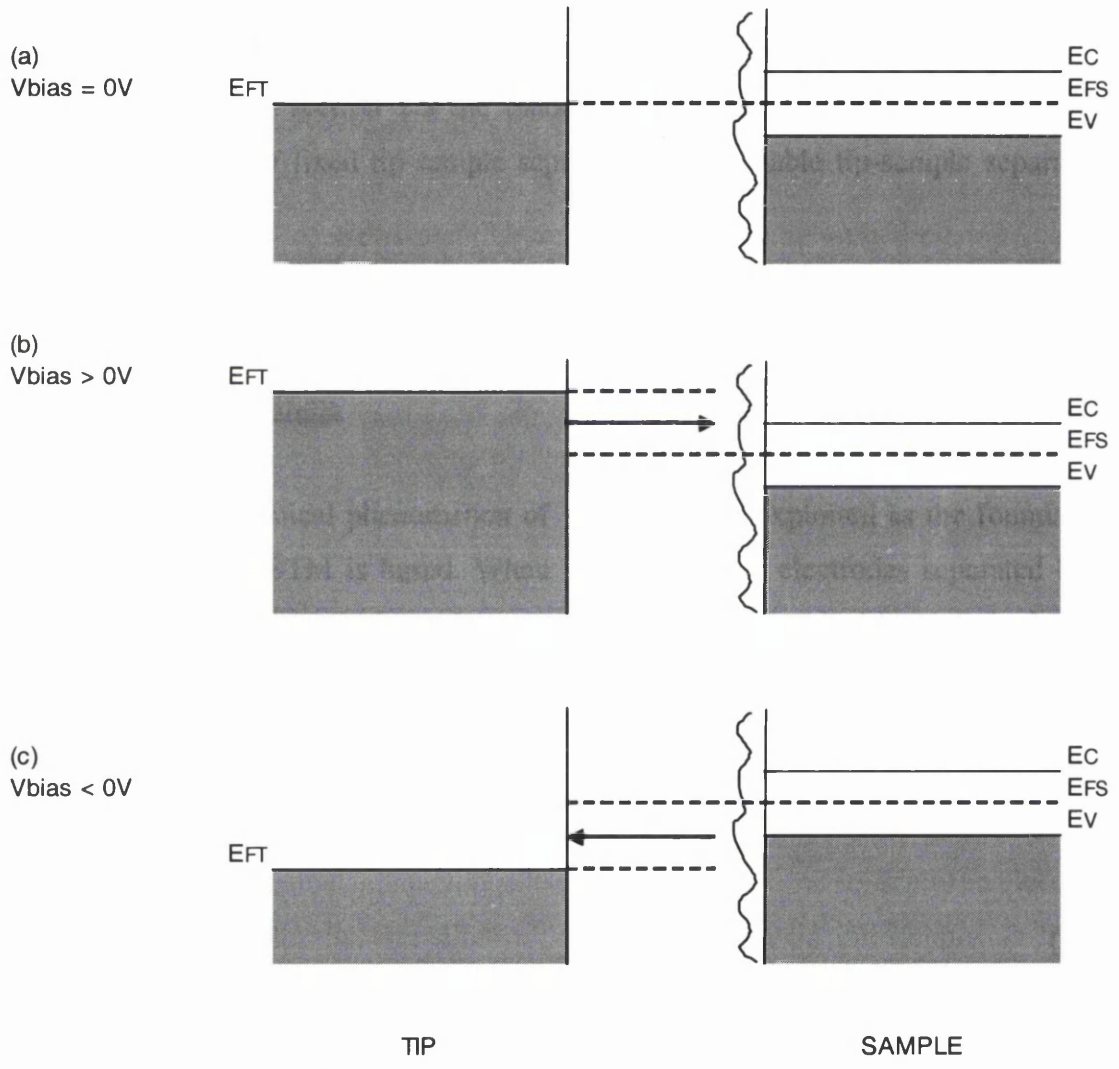


Figure 2.1.1: Schematic illustration of the tunnelling process between a metallic tip and semiconducting sample under differing bias conditions. (a) Tip and sample in thermal equilibrium separated by a vacuum barrier of appropriate width to promote tunnelling. (b) Under positive sample bias the net tunnel current arises from electrons tunnelling from the occupied states in the tip to the sample. (c) Under negative sample bias electrons tunnel from occupied states in the sample to the tip.

2.1.2 Principle of operation.

As discussed in section 2.1.1 the phenomena of quantum mechanical tunnelling is exploited by STM. A metallic tip, usually chemically etched from either Tungsten or Platinum wire and ideally possessing a single atom at its apex, is brought sufficiently close to a conductive sample to promote tunnelling.

Precise positioning of the tip in reference to the sample is achieved with piezoelectric drivers. The piezoelectric drivers are controlled by application of an electric field, allowing precise movements on the atomic scale. The tip is scanned laterally across the sample surface whilst a feedback loop maintains the tunnel current at preset value. Since the magnitude of the tunnel current is extremely sensitive to barrier width (equation 2.1) the STM is sensitive to small variations in sample topology and therefore possesses high vertical resolution. During the scan the voltage applied to the piezoelectric driver responsible for vertical movement is recorded and is representative of the sample topology. Figure 2.1.2 illustrates schematically the operating principle of the STM.

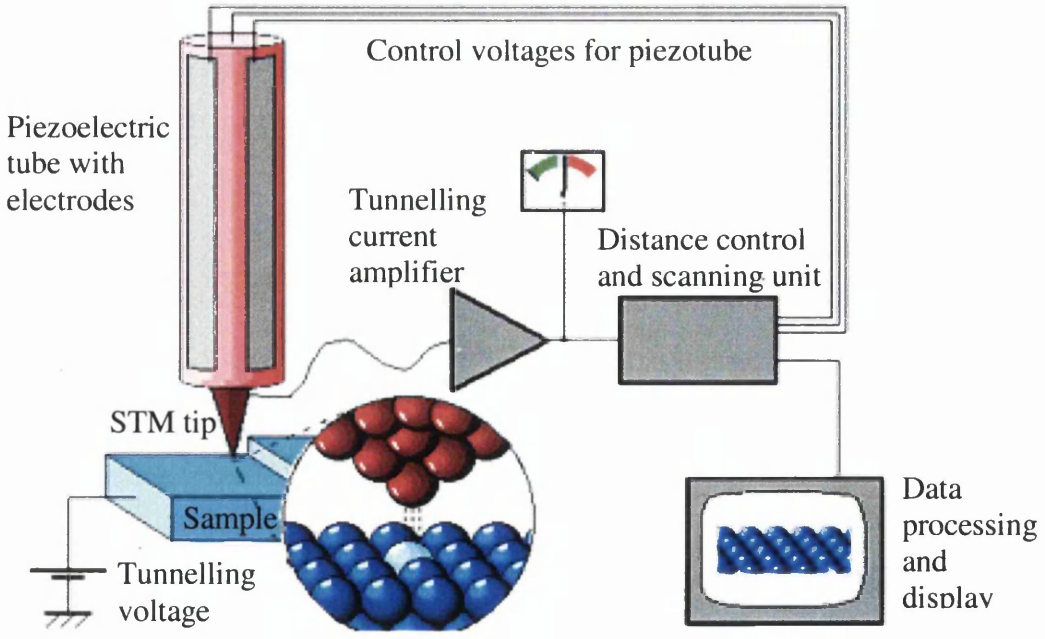


Figure 2.1.2: Schematic illustrating the operating principle of STM.

2.2 Scanning tunnelling microscopy (STM) theory.

2.2.1 Theory of STM.

For realistic models of the tip and sample surface, it is not feasible to calculate the transmission coefficient for an electron incident on the vacuum barrier. However, for the typical tip-sample separations encountered in STM (of the order 10Å, nucleus-to-nucleus) the coupling between tip and sample is weak and the tunnelling current may be treated with first order perturbation theory as proposed by Tersoff and Hamann[4]. In first order perturbation theory the tunnel current can be expressed by equation 2.2.

$$I = \frac{2\pi}{\hbar}\sum_{\mu,\nu} f(E_\mu)[1 - f(E_\nu + eV)]\left|M_{\mu\nu}\right|^2 \delta(E_\mu - E_\nu) \qquad (2.2)$$

Where f(E) is the Fermi function, V is the applied potential , $M_{\mu\nu}$ is the tunnelling matrix element between states ψ_μ of the tip and ψ_ν of the sample surface. E_μ and E_ν are the energies of ψ_μ and ψ_ν respectively in the absence of tunnelling. The contribution of reverse tunnelling has been neglected as it is only apparent at higher temperatures. The tunnelling matrix element can be expressed by equation 2.3.

$$M_{\mu\nu} = \frac{\hbar^2}{2m}\int dS.(\psi_\mu^* \nabla\psi_\nu - \psi_\nu \nabla\psi_\mu^*) \qquad (2.3)$$

Where the integral is over any surface lying entirely within the barrier region. To calculate the tunnel current a model for the tip has to be adopted, since calculations based on the physical atomic structure of the tip are not feasible. The model adopted for the tip is mathematical point source of current since this choice maximises resolution whilst removing the necessity to consider complex tip-sample interactions. By using a mathematical point source of current to represent the tip, equation 2.2 reduces to equation 2.4. Equation 2.4 suggests that an ideal STM would measure $\rho(r_t,E_F)$, which is the local density of states (LDOS) at the Fermi level for the bare surface at the position of the tip.

$$I \propto \sum_\nu \left|\psi_\nu(r_t)\right|^2 \delta(E_\nu - E_F) \equiv \rho(r_t, E_F) \qquad (2.4)$$

Therefore this model interprets STM results as measuring the property of the bare surface without reference to the complex tip-sample interactions.

2.2.2 Operating modes.

The most commonly employed operating modes for STM are constant current and constant height [2], [5] modes. Details regarding the operation of these modes are discussed below.

Constant current

In constant current operation a feedback loop continually adjusts the tip-sample separation in order to maintain the tunnel current at a preset value. The tip is scanned in two directions laterally across the sample surface and the voltage applied to the piezoelectric driver in order to maintain constant current is recorded as a function of position. Topographic information is inferred from the recorded voltage as a function of position. Due to the function of the feedback loop, constant current operation is applicable to surfaces which are not atomically flat. However, the scan speed of constant current operation is lower than constant height operation due to the finite response time of the feedback loop.

Constant height

In constant height operation the feedback loop is either disabled or slowed to the point where the feedback no longer adjusts tip sample separation in response to local variations in sample topography. The tip is scanned in two dimensions laterally across the surface of the sample at a constant height and the tunnel current as a function of position recorded. Topographic information is yielded directly by the recorded tunnel current as a function of position. This operational mode offers faster scan rate since it is not encumbered by the response time of the feedback loop, therefore image distortion due to thermal drift and piezoelectric hysteresis may be significantly reduced. Constant height operation is only applicable to atomically flat surfaces, since adjustments for sample roughness are not made.

2.2.3 STM application to Si(111) 7x7 and GaAs(110) surfaces.

STM of Si(111) 7x7

The 7x7 reconstruction of the Si(111) surface has been extensively studied using various techniques, first imaged successfully by Binning and Rohrer [6] using STM in 1984. Binning and Rohrer's observations of the 12 adatoms present within the 7x7 reconstruction provided important information which aided Takayanagi *et al* [7] to produce the widely accepted dimer-adatom-stacking fault (DAS) model illustrated by figure 2.2.1.

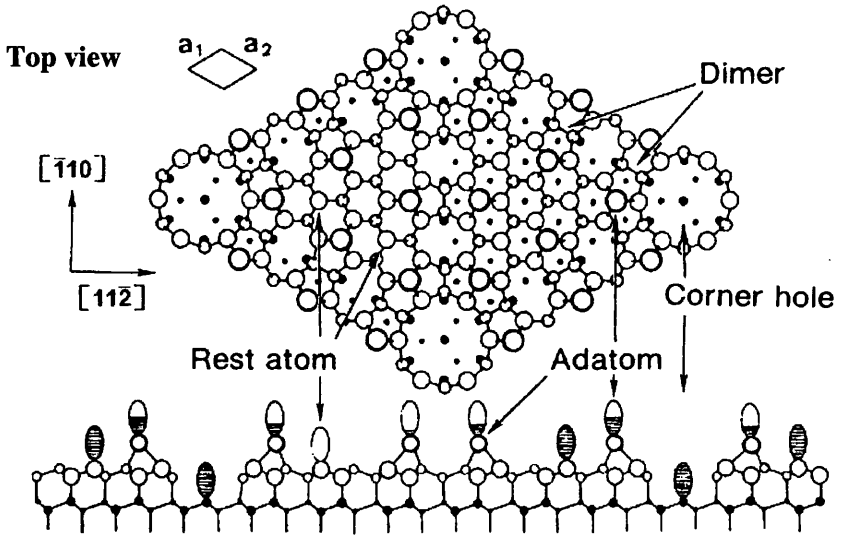


Figure 2.2.1: The dimer-adatom-stacking fault (DAS) model of the Si (111) 7x7 reconstruction. Taken from [7].

Figure 2.2.2 shows an STM image of a sample prepared under UHV conditions to yield an atomically clean Si(111) surface, illustrating the 7x7 reconstruction in which the bright and dark regions indicate protrusions and depressions within the image respectively. The diamond shape marked whose corners are located at depressions in the image represents the 7x7 unit mesh. The 12 protrusions present within the unit mesh of the 7x7 reconstruction are the positions of the 12 adatoms in the DAS.

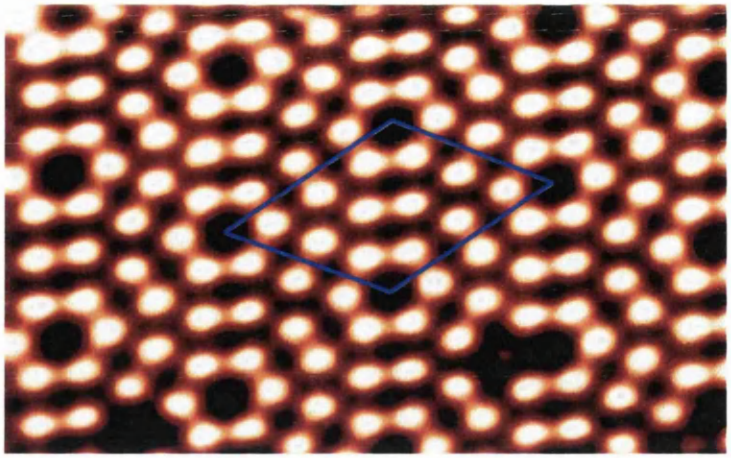


Figure 2.2.2: STM image of the 7x7 reconstruction of Si(111). The diamond shape marked illustrates the 7x7 unit mesh.

STM of GaAs(110)

Of the III-V compound semiconductors, research has been predominantly directed toward GaAs, with study focussing on the (110) surface which may be prepared by cleavage in UHV, to yield an atomically clean surface. The crystalline structure of GaAs is of the Zincblende type [8], which possesses fcc translational symmetry with a two atom basis. A Ga atom is positioned at (0,0,0) and an As atom at (¼, ¼, ¼) of the non primitive fcc unit cube as illustrated by figure 2.2.3.

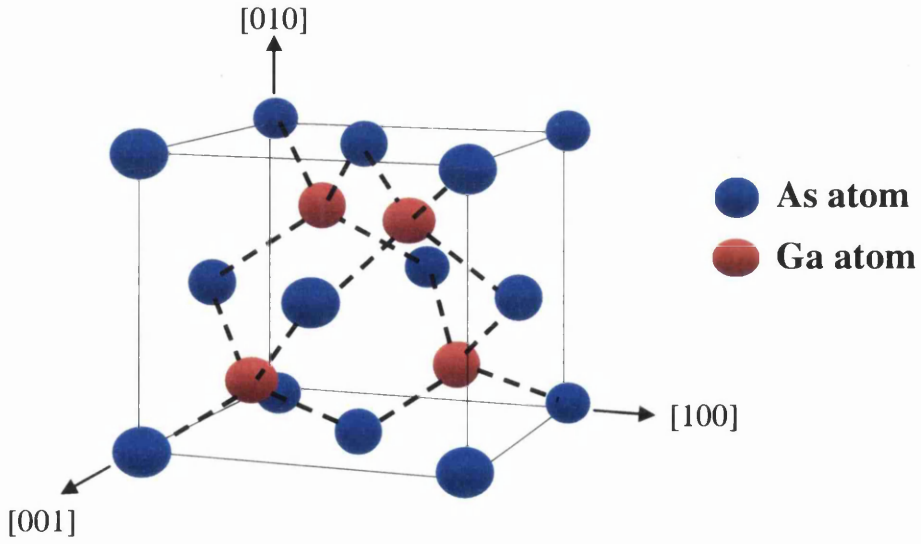


Figure 2.2.3: Schematic representation of the unit cube for GaAs.

Cleavage of the GaAs crystal along the (110) family of planes exposes a non-polar surface and creates two dangling bonds per unit cell. The surface is termed non-polar because it possesses equal numbers of Ga and As atoms. Following cleavage charge transfer occurs between Ga and As species on the surface, leading to the Ga species becoming electropositive and the As species electronegative. Moreover, the surface relaxes to attain an energetically favourable state resulting in surface buckling as the As species move outward from the surface whilst Ga move inward; this is illustrated by figure 2.2.4 and can be characterised by a rotation angle ω.

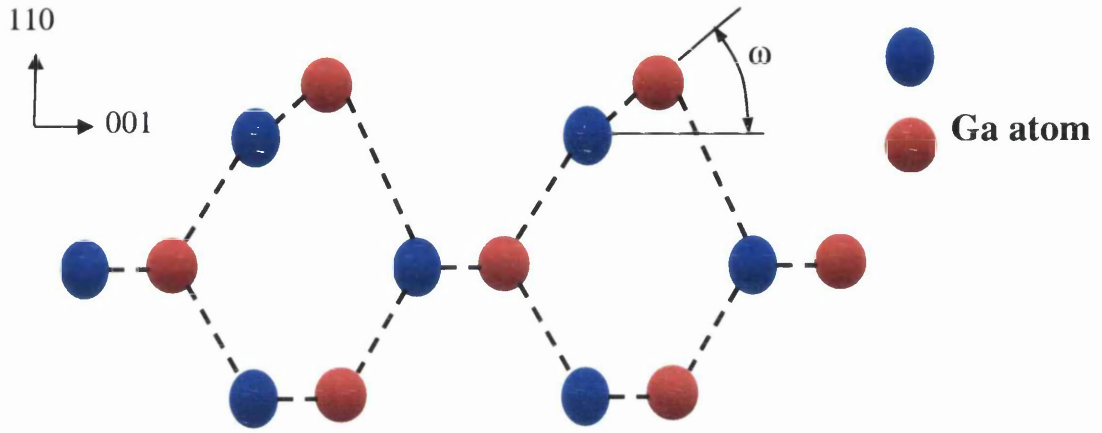


Figure 2.2.4: Illustration of the buckling of the surface of GaAs(110) following relaxation of the surface atoms.

STM images obtained at positive sample bias reveal the position of unoccupied states concentrated on Ga atoms, whilst images obtained at negative sample bias reveal the position of occupied states concentrated on As atoms. STM images for both negative and positive bias conditions are shown in figure 2.2.5, along with a model illustrating the positions of the Ga and As atoms in the images.

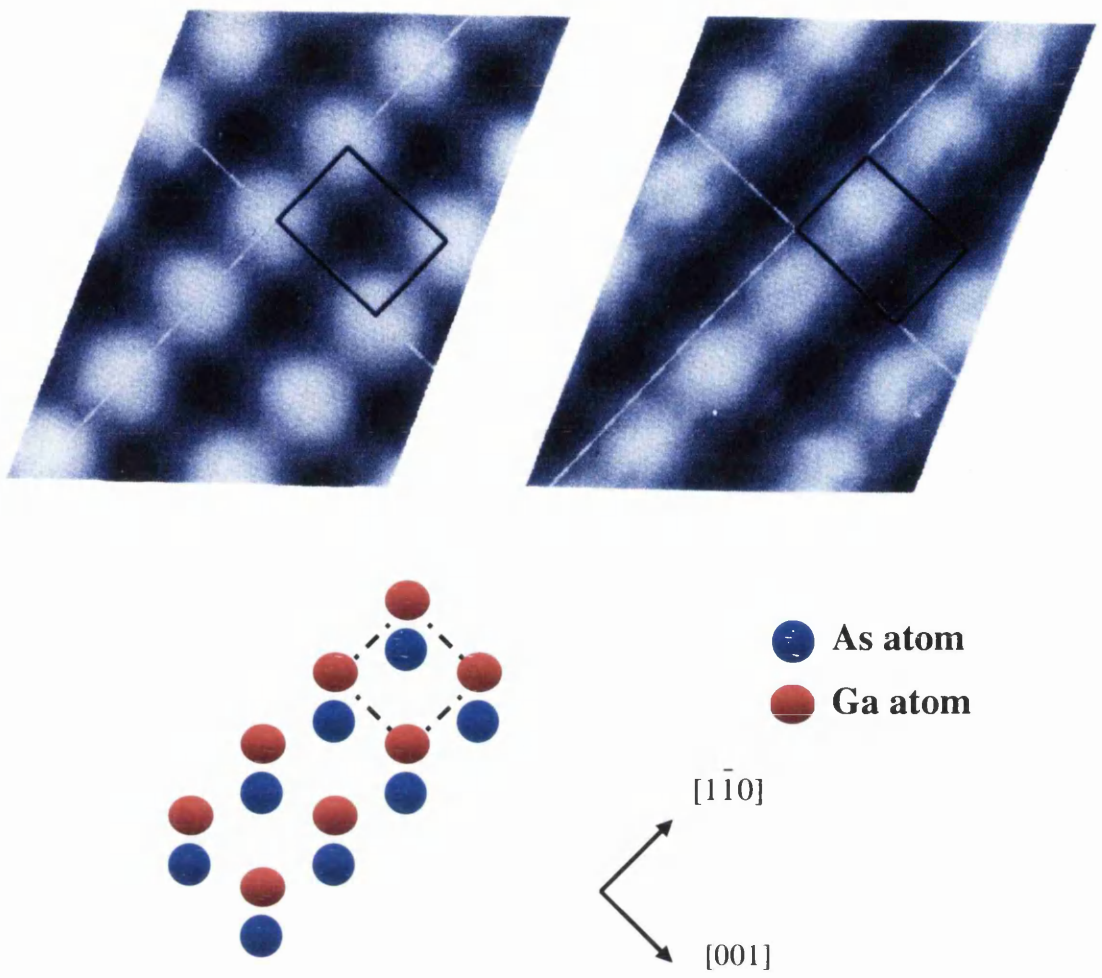


Figure 2.2.5: STM images of the surface of GaAs(110) under (a) negative bias and (b) positive bias conditions along with (c) a model of the surface illustrating the positions of Ga and As atoms. Obtained from[].

2.3 Scanning tunnelling spectroscopy (STS).

2.3.1 Theory of STS.

Under the simplifying assumptions that: the tip possess a uniform density of states, the applied bias does not exceed the work functions of either the tip or sample surface, the operating temperature is low and only s-wave tip wave functions are important; the Tersoff-Hamann model predicts that the tunnel current will take the form of equation 2.5.

$$I \approx \int_{E_F}^{E_F+V} \rho(r_t, E)dE \tag{2.5}$$

Equation 2.5 neglects the energy and potential dependence of the matrix elements and the tip density of states. To account for the voltage dependence of the surface wave functions an additional barrier transmission coefficient T(E,V) is included in equation 2.5, yielding equation 2.6.

$$I \propto \int_{E_F}^{E_F+V} \rho(E)T(E,V)dE \tag{2.6}$$

In practice bias voltages of several volts are typically used to study semiconductors, which violates the low voltage assumption made by the Tersoff-Hamann model. In this situation the simple planar models using the Wentzel-Kramers-Brillouin (WKB) approximation can be applied and the tunnel current is given by equation 2.7.

$$I = \int_{0}^{eV} \rho_s(r,E)\rho_t(r,-eV+E)T(E,eV,r)dE \tag{2.7}$$

Where $\rho_t(r,E)$ and $\rho_s(r,E)$ are the density of states for the tip and sample surface at location r and energy E respectively, measured with reference to the individual Fermi levels of both the tip and sample surface. T(E,eV,r) is the tunnelling transmission probability which is given by equation 2.8.

$$T(E,eV) = \exp\left(\frac{2Z\sqrt{2m}}{\hbar}\sqrt{\frac{\phi_s+\phi_t}{2}+\frac{eV}{2}-E}\right) \tag{2.8}$$

Where Z is the tip-sample separation, ϕ_t and ϕ_s are the work functions of the tip and sample surface respectively.

The differential conductivity dI/dV as shown in equation 2.9 yields a quantity that relates to the density of states but without a simple relationship. Feenstra *et al* [9] demonstrated that by normalising the differential conductivity dI/dV by the total conductivity I/V provides a quantity corresponding to the normalised density of states.

$$dI/dV \propto e\rho(eV)T(eV,eV) + e\int_0^{eV}\rho(E)\frac{d}{d(eV)}[T(E,eV)]dE_V)]dE \tag{2.9}$$

The normalised conductivity is given by equation 2.10.

$$\frac{dI/dV}{I/V} = \frac{\rho(eV) + \int_0^{eV}\frac{\rho(E)}{T(eV,eV)}\frac{d}{d(eV)}[T(E,eV)]dE}{\frac{1}{eV}\int_0^{eV}\rho(E)\frac{T(E,eV)}{T(eV,eV)}dE} \tag{2.10}$$

Feenstra *et al* [9] has argued that since transmission probability terms T(eV,eV) and T(E,eV) appear as ratios in the second term of the numerator and denominator of equation 2.10 that their dependence on the separation and applied bias tend to cancel. Therefore the normalisation reduces to equation 2.11 which is representative of the normalised surface density of states.

$$\frac{dI/dV}{I/V} \equiv \frac{d\ln I}{d\ln V} \approx \frac{\rho(E)}{\frac{1}{eV}\int_0^{eV}\rho(E)dE} \tag{2.11}$$

For semiconductors with band gaps >0.5eV the aforementioned normalisation procedure breaks down because the ratio of differential conductivity dI/dV to the total conductivity I/V tends to diverge at the band edges. Feenstra proposed an alternative method of normalisation that is applicable to materials with larger band gaps [10]. The method convolves experimental I/V data with a suitable function to eliminate zeros within the band gap.

2.3.2 Fixed tip sample separation measurements.

To acquire fixed tip-sample separation spectroscopic information whilst imaging, the lateral and vertical movement of the tip is suspended for a period of time, during which the bias voltage is ramped in discrete steps over a defined range and the tunnel current recorded as a function of the bias voltage. Figure 2.3.1 schematically illustrates the acquisition of a single spectra with fixed tip-sample separation.

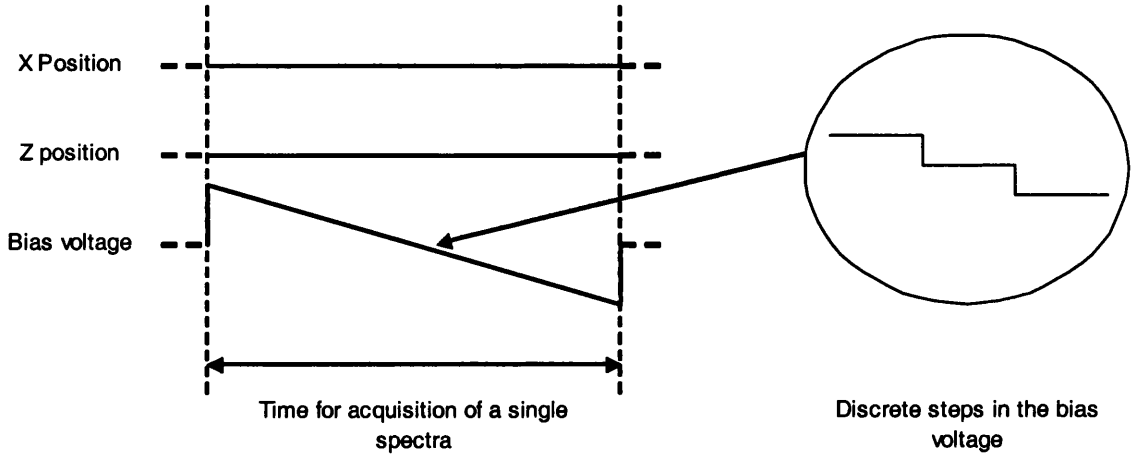


Figure 2.3.1: schematic illustration of the acquisition of a single fixed tip-sample separation spectra.

To prevent the feedback loop compensating for variations in the tunnel current brought about by changing the bias voltage, it is disabled during spectroscopic measurements.

Conventionally the differential conductivity is obtained by differentiating the measured tunnel current (equation 2.9). An alternative method to obtain the differential conductivity is by direct measurement with a lock-in amplifier [11]. To obtain the differential conductivity, a sinusoidal modulation (typically 1KHz) whose frequency is greater then the cut off frequency of the feedback loop is superimposed onto the bias voltage, and the differential conductivity calculated by the lock-in amplifier from the resulting modulation in the tunnel current.

2.3.3 Variable tip sample separation measurements.

When performing spectroscopic measurements on semiconducting materials with fixed tip sample separation, the tunnel current can range over many orders of magnitude due to the presence of band gaps in the material, therefore resulting in the inability to accurately resolve band edges and states within the band gap.

To improve the dynamic range of the measurement many spectras at different discrete fixed tip-sample separations can be obtained at the same location. However, since spectras are acquired in a piecewise fashion using this method, significant processing is required to extract the final complete spectra. Alternatively, Feenstra proposed

manipulating the tip-sample separation during spectral acquisition to amplify the tunnel current [12], [13], [14]. This is achieved by continuously varying the tip-sample separation during the measurement and mathematically normalising the spectra obtained to remove the effects of varying the tip-sample separation.

As the bias voltage is applied in discrete steps over a defined range the tip-sample separation is varied as a function of the bias voltage, such that s=s(V). For each discrete step in bias voltage the tunnel current I_m [s(V),V] and differential conductivity σ_m [s(V),V] are measured. The differential conductivity, which is the derivative of the current with respect to voltage at a particular value of tip-sample separation as described by equation 2.12 is obtained using a lock-in amplifier.

$$\sigma_m[s(V),V] \equiv \left.\frac{\partial I_m}{\partial V}\right|_s \tag{2.12}$$

The simplest form of tip-sample separation contour is a linear ramp, where the tip is linearly moved toward the sample surface as the magnitude of the bias voltage is decreased, and then retracted from the sample surface as the magnitude of the bias voltage is increased. Thus the form of the contour is given by equation 2.13.

$$\Delta s(V) = s(V) - s_0 = +a|V| \tag{2.13}$$

Where s_0 is the tip-sample separation at a bias voltage of zero volts, and a determines the slope of the ramp. Feenstra suggests that a value for a in the range 0.5 to 2Å/V is suitable for general measurements. Figure 2.3.2 illustrates the application of variable tip-sample technique by Feenstra to a monolayer of Sb on a GaAs(110) substrate [12]. A more sophisticated choice of tip-sample contour based on the magnitude of the tunnel current at a particular bias voltage may be applied where the linear ramp does not yield sufficient improvement in the dynamic range of the measurement.

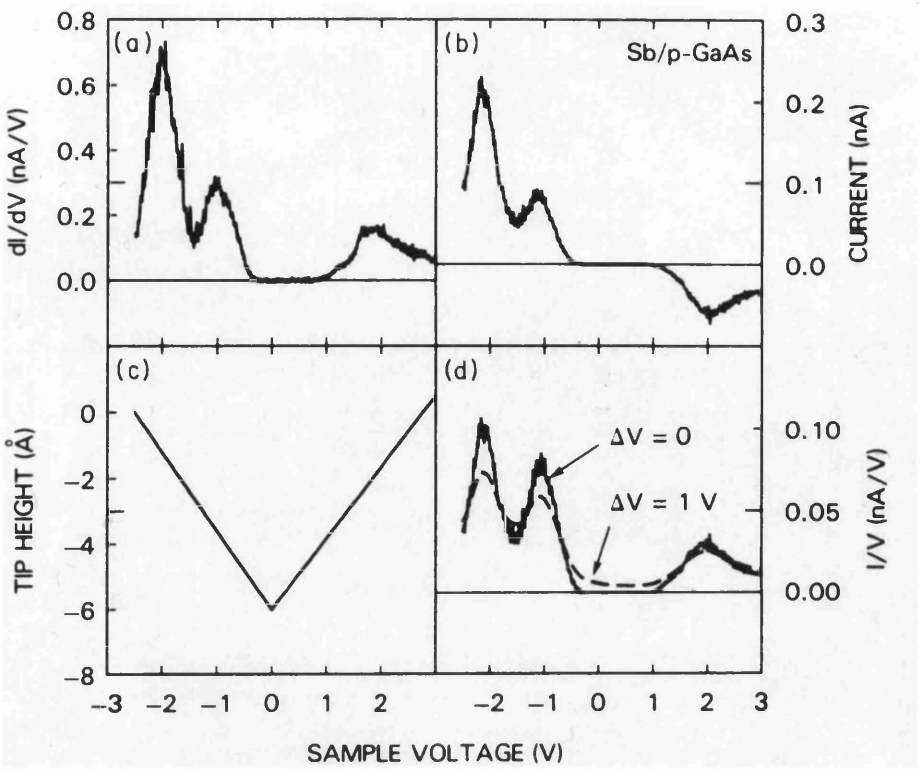


Figure 2.3.2: Raw data for (a) the differential conductivity, (b) the current as a function of bias voltage obtained from an ordered region of a monolayer of Sb on GaAs(110) and (c) the variation of tip-sample separation as a function of the applied bias and (d) the total conductivity. Taken from [12].

Transformation to fixed tip-sample separation equivalency

To transform spectroscopic data acquired by the variable tip-sample separation method to constant separation, Feenstra proposes a method based on the use of the logarithmic derivative [13]. The logarithmic derivative $g_m[s(V),V]$ as described by equation 2.14 is the ratio of the differential conductivity to the tunnel current and is approximately independent of the tip-sample separation for small changes in separation.

$$g_m[s(V),V] \equiv \frac{\sigma_m[s(V),V]}{I_m[s(V),V]} \tag{2.14}$$

Denoting the logarithmic derivative at some constant value of s = s' to be g(s',V) then we have equation 2.15.

$$g(s',V) \approx g_m[s(V),V] \tag{2.15}$$

From the definition of g(s',V) we have equation 2.16

$$g(s',V) \equiv \frac{\sigma(s',V)}{I(s',V)} = \frac{d}{dV}\ln I(s',V) \tag{2.16}$$

The derivative on the right hand side of equation 2.16 is only valid because the current is evaluated at constant s = s'. Equation 2.17 follows from equation 2.16 for any V and V' with the same sign.

$$\frac{I(s',V)}{I(s',V')} = \exp\left\{\int_{V'}^{V} g(s',E)dE\right\} \tag{2.17}$$

Differentiating equation 2.17 with respect to V yields equation 2.18, which is the conductivity at any value of s, in particular at s = s'. The choice of V' is arbitrary.

$$\sigma(s',V) = I(s',V')g(s',V)\exp\left\{\int_{V'}^{V} g(s',E)dE\right\} \tag{2.18}$$

To evaluate the conductivity at constant tip-sample separation in terms of measured quantities equation 2.19 is used, which is evaluated separately for positive and negative voltages.

$$\sigma(s',V) \approx I_m[s(V'),V']g_m[s(V),V]\exp\left\{\int_{V'}^{V} g_m[s(E),E]dE\right\} \tag{2.19}$$

Normalisation of conductivity data

As stated previously, for materials with surface band gaps in excess of 0.5eV, taking the ratio of differential conductivity (σ_m) to total conductivity (I_m/V) in order to normalise the spectroscopic data is inappropriate [12], since the ratio diverges at the band edges. This divergence is due to both the tunnel current and conductivity

approaching zero at the band edges. The analysis method breaks down because I/V is no longer a valid estimation of the tunnelling transmission term which is non zero in the band gap region.

To overcome this problem several methods have been proposed by various authors. Martensson *et al* [14] suggests broadening I/V by convolution with a suitable function, thereby eliminating zeros of the I/V within the band gap region. Suitable choices of convolving functions were exponential or Gaussian. An alternative method to eliminate zeros in the band gap region was proposed by Prietsch *et al* [15]. An offsetting constant of suitable size (typically c = 0.02 nA/V) is combined with the I/V data as shown in equation 2.20.

$$\frac{\overline{I}}{V} = \sqrt{\left(\frac{I}{V}\right)^2 + c^2} \qquad (2.20)$$

2.3.4 STS measurements of Sb on GaAs(110).

Here a presentation of work carried out by Feenstra [16], [17] is undertaken, illustrating the application of variable tip-sample separation STS measurements to an Sb overlayer on GaAs(110). Metal adsorbates such as Sb introduce states within the bandgap, resulting in pinning of the Fermi level. The improved dynamic measurement range associated with variable tip-sample separation STS is exploited here to reveal states within the bandgap that pin the Fermi level. A GaAs(110) substrate is employed because dangling bonds produced by the cleaving process contribute surface states outside the bulk bandgap, therefore allowing investigation of states introduced into the band gap region by an Sb overlayer.

Feenstra deposited Sb onto highly doped (1-4 x 10^{18} cm^{-3}) p-type GaAs(110) surfaces with various coverages ranging from a fraction of a monolayer (1 monolayer is equivalent to 8.85 x 10^{14} atoms/cm^2) to 1 monolayer. STM and STS were performed concurrently to allow registry between topographic information and STS spectra. In order to ensure registry between STM images and variable tip-sample separation STS spectras, the residual thermal drift was less than 0.1Å/s. Figure 2.3.3 shows an 120 x 120 Å STM image of Sb on GaAs(110).

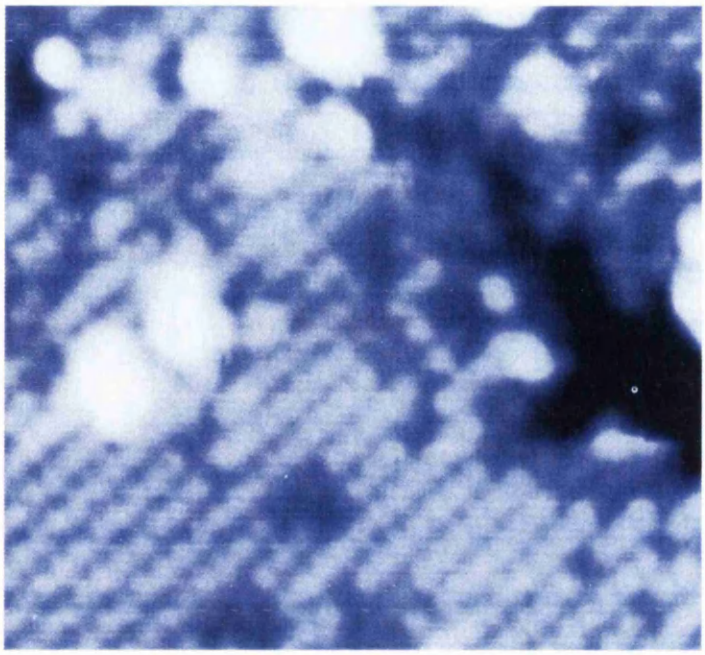


Figure 2.3.3: An 120 x 120 Å STM image of Sb on GaAs(110) taken from [17].

Sb forms flat ordered terraces with Sb atoms occupying positions close to those expected by extension of the bulk. Figure 2.3.4 shows a model illustrating the positions of Sb in reference to the bulk Ga and As atoms for GaAs(110).

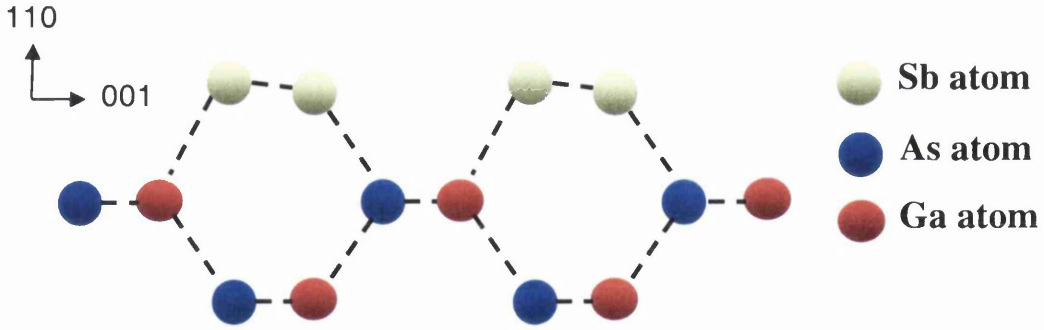


Figure 2.3.4: A model illustrating the position of Sb atoms in reference to the bulk Ga and As atoms for GaAs(110).

Spectroscopy measurements were performed at locations illustrated by markers in figure 2.3.5. Spectra (c) and (d) were obtained from well ordered sections of the Sb terraces, which possess bangaps with normalised conductivities close to zero and with

similar magnitudes to those of the GaAs substrate. Spectra (a) and (b) were obtained from terrace edges and indicate an intense state at approximately 0.8 V within the bandgap region causing pinning of the Fermi level. This highlights the power of variable tip-sample separation STS as a method to gain valuable information regarding the detailed electronic properties of surfaces and interfaces on the nanoscale.

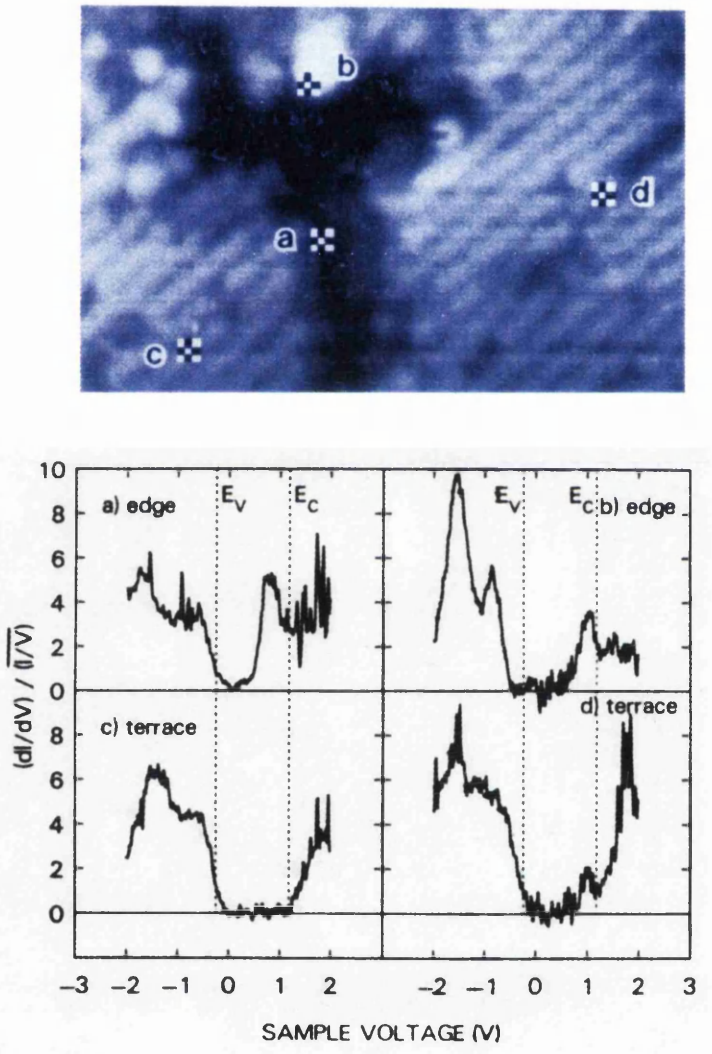


Figure 2.3.5: Normalised conductivity versus voltage measured at locations indicted by markers for an Sb overlayer on GaAs(110). Valence band and conduction band edges are marked Ev and Ec respectively. Taken form [17].

2.4 References.

[1] S.O. Kasap, 'Principles of Electrical Engineering Materials and Devices', Chapter 3, **183 – 188**, McGraw Hill (1997).
[2] J. Tersoff, N.D. Lang, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 1, **1 – 3**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[3] R.J. Hamers, 'Scanning Tuneling Microscopy 1', Chapter 5, **83 - 85**, Edit R.Gomer, Second edition, Springer-Verlag (1994).

[4] J. Tersoff and D.R. Hamaan, Phys. Rev. B **31**(2), 805 (1985).

[5] R. Wiesendanger, H.J. Guntherodt, 'Scanning Tuneling Microscopy 1', Chapter 1, **5 - 8**, Edit R.Gomer, Second edition, Springer-Verlag (1994).

[6] G. Binning, H. Rohrer, Ch. Gerber, and E. Weibel, Phys. Rev. Lett, **50**, 120 (1983).

[7] K.Takayanagi, Y. Tanishiro, M. Takahashi and S. Takahashi, J. Vac. Sci Technol, A **3**, 1502 (1985).

[8] S.O. Kasap, 'Principles of Electrical Engineering Materials and Devices', Chapter 1, **42**, McGraw Hill (1997).

[9] R.M. Feenstra, J.A. Stroscio and A.P.Fein, Surf. Sci, **181**, 295(1987)

[10] J.A. Stroscio, R.M. Feenstra, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 4, **138 - 141**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[11] R.J. Hamers, D.F. Padowitz, 'Scanning Probe Microscopy and Spectroscopy Theory, Techniques, and Applicatons', Chapter 4, **73**, Edit D. Bonnell, Second edition, Wiley-VCH (2001).

[12] J.A. Stroscio, R.M. Feenstra, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 4, **134 - 138**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[13] C.K. Shih, R.M. Feenstra and G.V.Chandrashekhar, Phys. Rev. B **43**, 7931 (1991).

[14] P. Martensson and R.M. Feenstra, Phys. Rev. B **39**, 7744 (1989).

[15] M. Prietsch, A. Samsavar and R. Ludeke, Phys. Rev. B **43**, 11850 (1991).

[16] R.M. Feenstra and P. Martenesson, Phys. Rev. Lett. **61**, 447 (1988).

[17] J. Tersoff, N.D. Lang, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 5, **251** – **275**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

Chapter 3

Experimental Techniques

Introduction

Herein a brief description of the Omicron Micro-STM and STM/SEM-HC microscopes and associated SCALA control electronics is undertaken. Additionally, an overview of the SCALA Pro software utilised by both microscopes and its application in making topographic and spectroscopic measurements is presented.

A comprehensive description of the external connections the SCALA electronics provide and the nano-structuring language inherent within the SCALA Pro software are given, due to their extensive usage during the latter development of instrumentation used to implement variable tip-sample separation.

Lastly, the electrochemical etching method used to prepare tips for both microscopes is presented.

3.1 The Omicron UHV Micro-STM and STM/SEM-HC systems.

3.1.1 The Omicron UHV Micro-STM

The Omicron UHV Micro-STM system is fixed to an 8" OD flange and contained within a custom designed chamber mounted on a VG ESCAlab MKII as illustrated by figure 3.1.1.

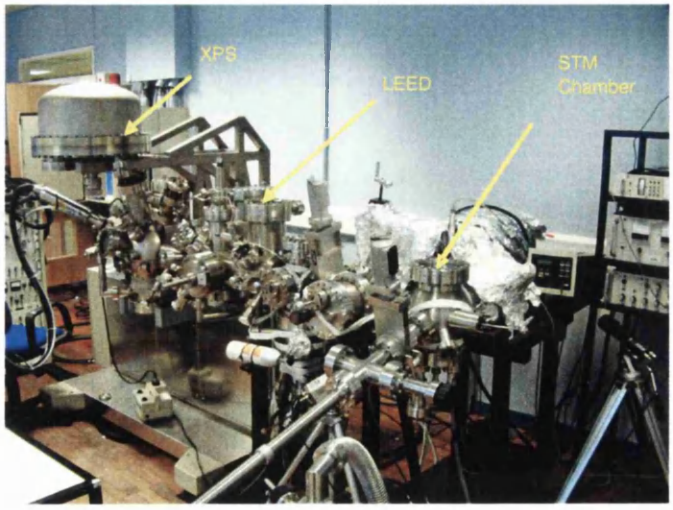


Figure 3.1.1: A photograph of the VG ESCAlab MKII incorporating the Omicron Micro-STM system.

The microscope unit as shown in figure 3.1.2 consists of the STM head comprising a tube type tip scanner and sample stage.

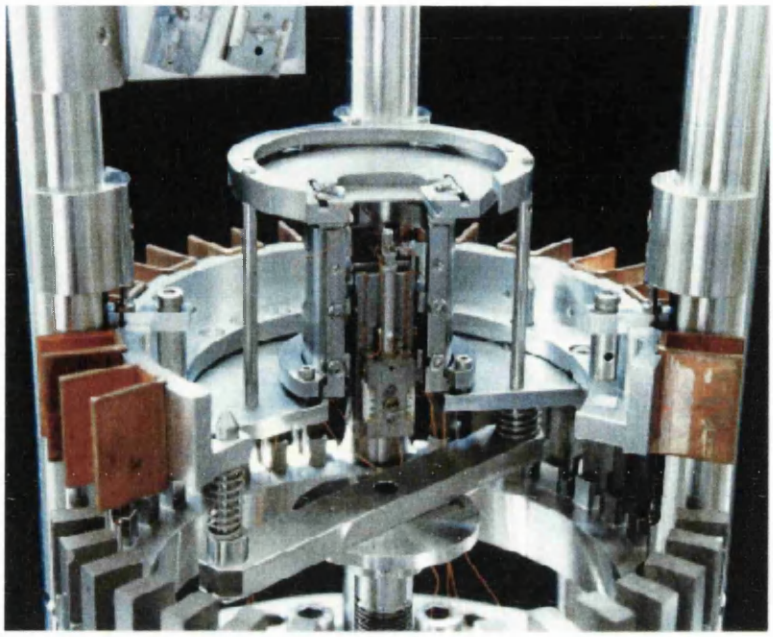


Figure 3.1.2: A photograph of the Omicron Micro-STM system whilst parked.

The tube type scanner is manufactured from piezoelectric ceramic [1] and facilitates precise spatial control of tip position on the atomic scale. The Lateral resolution (x, y) of the scanner is of the order of 1Å whilst vertical (z) resolution is of the order of 0.05Å. During operation of the microscope a bias voltage is applied between the sample and tip. In this case, the bias voltage is applied to the sample whilst the tip is at ground potential. Copper fins situated at regular intervals around the periphery of the microscope unit act in conjunction with permanent magnets to allow the microscope unit to levitate, providing vibration isolation from the external environment. Further damping is provided by three springs on to which the microscope unit is lowered during operation.

3.1.2 The Omicron STM/SEM-HC

The Omicron STM/SEM-HC system combines STM and scanning electron microscopy (SEM) techniques, which may be operated simultaneously to allow precise positioning of the tip on a sample. The STM microscope unit is mounted on an

8" OD flange and housed within a UHV chamber. Figures 3.1.3 and 3.1.4 show photographs of the STM microscope unit and complete multi-chambered UHV system providing a multi-technique platform.

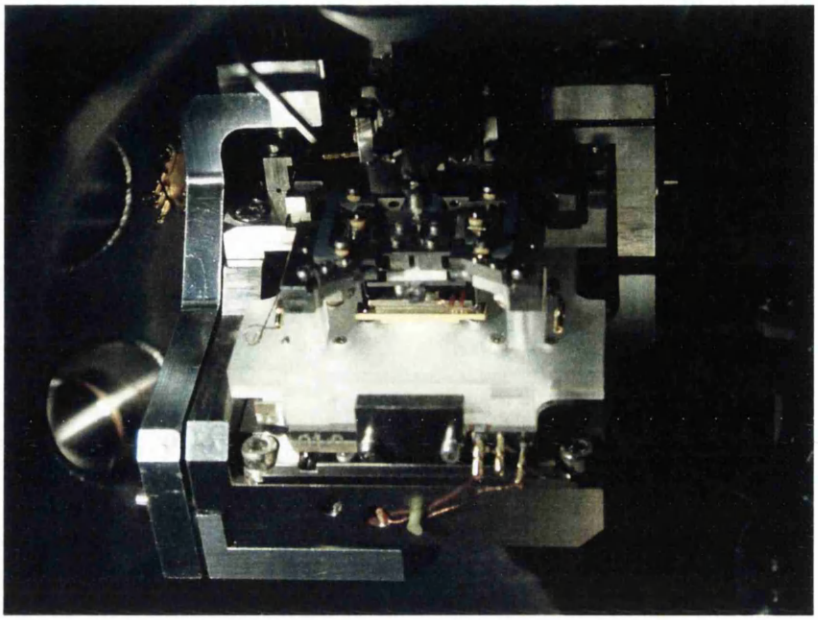


Figure 3.1.3: A photograph of the STM microscope unit.

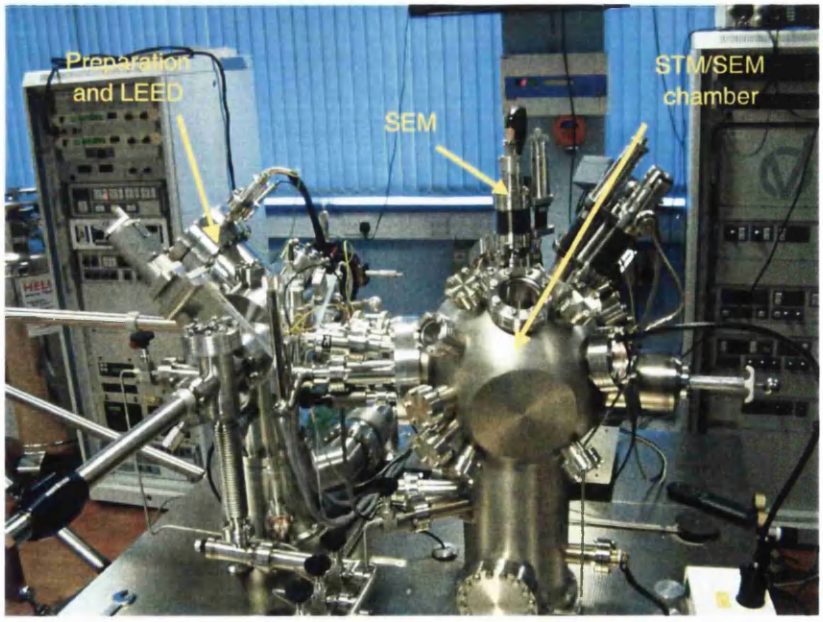


Figure 3.1.4: A photograph of the complete system incorporating STM, SEM and LEED analysis techniques.

The sample may be either cooled or heated in-situ. Sample cooling is provided by a LHe cryostat, allowing the sample to be cooled to within sixty degrees of absolute zero. Both radiative and direct current heating techniques are provided to allow the sample to be heated.

Coarse positioning of the tip is achieved by piezo inertia drives, which provide independent control of tip position in three special directions: 5mm in X, 12mm in Y and 5mm in Z. A tube type scanner similar to the one described previously allows fine movement of the tip in three special directions. Further, piezo inertia drives allow 10mm x 10mm orthogonal movement of the sample stage. During operation of the microscope a bias voltage is applied between the sample and tip. In this case, the bias voltage is applied to the tip whilst the sample is at ground potential.

Vibration isolation is provided by two means: the microscope stage is damped using Viton rings and overall damping is provided by air legs which levitate the UHV chamber.

3.2 The Omicron SCALA.

Here a brief description of the operation of the SCALA electronics and SCALA Pro software as utilised by both the aforementioned microscopes is undertaken. Further information regarding any aspects of the operation of the SCALA electronics or SCALA Pro software discussed here can be obtained from the appropriate reference manuals [2], [3].

The SCALA electronics are contained within a 19" rack mounting case and connect to a host PC via a custom PCI interface card, which allows parameters relating to the operation of the STM system to be altered on a real time basis. Figure 3.2.1 shows a picture of the SCALA electronics connected to a host PC system.

Before measurements can commence the tip requires positioning within the area of interest on the sample. Provision is made within the SCALA electronics for coarse positioning of both the tip scanner and sample stage via piezo inertia drives. An

external user interface pictured in figure 3.2.2 allows direct control of the coarse piezo inertia drive and therefore the coarse position of the tip in relation to the sample.

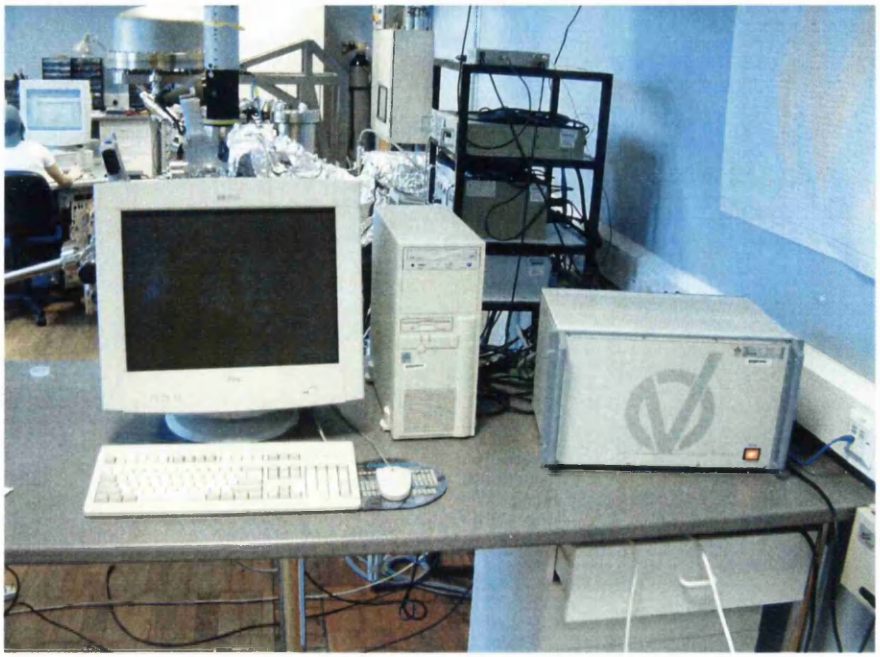


Figure 3.2.1: A picture showing the SCALA electronics connected to a host PC.

Figure 3.2.2: A picture showing the user interface provided by the SCALA electronics to facilitate tip positioning.

To achieve tunnelling the tip needs to be brought within a few angstroms of the surface. The tip is coarsely positioned visually with the aid of video cameras and then an automatic approach initiated. The tip-sample separation is reduced until a preset tunnel current is detected, at which point approach is terminated. The automatic approach process is described in figure 3.2.3.

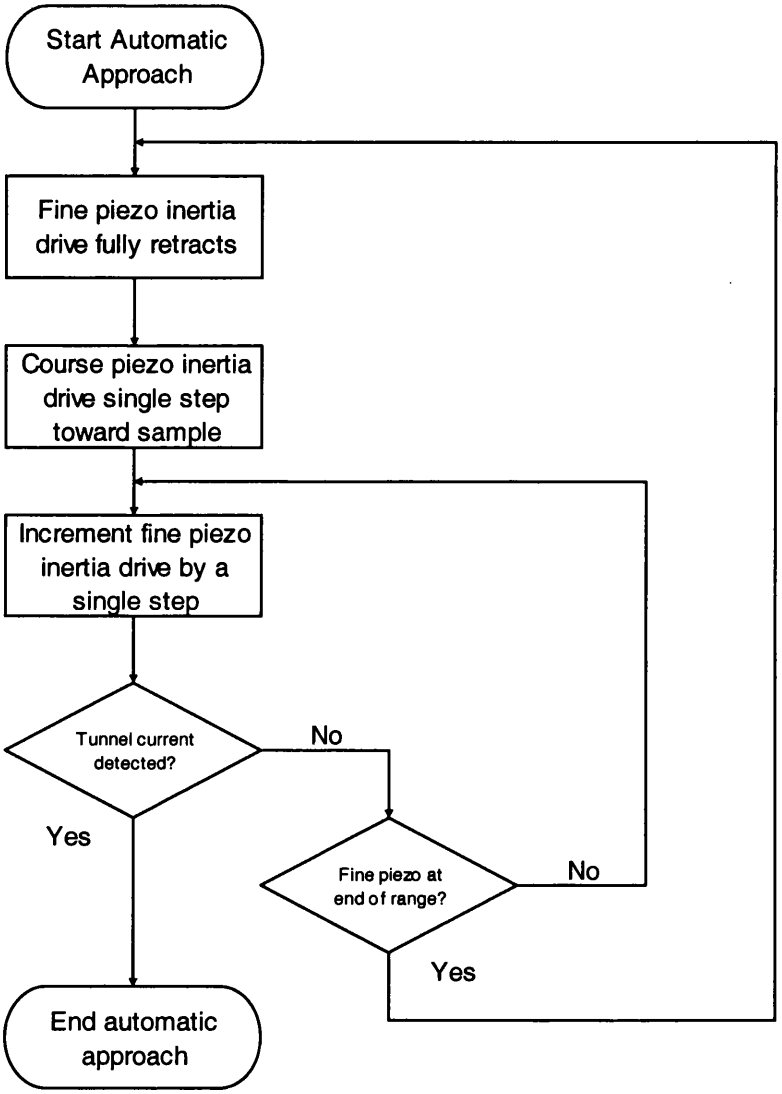


Figure 3.2.3: A flow chart describing the automatic approach process implemented by the SCALA electronics to allow the tip to be brought within the tunnelling regime.

Initially the Z-piezo inertia drive responsible for fine tip positioning is fully retracted. A single coarse step toward the sample is performed by the coarse Z-piezo inertia drive. The fine piezo incrementally steps the tip toward the sample through its complete range. During this process if the preset tunnel current is detected the

approach is terminated, otherwise the fine piezo is completely retracted and the process repeated until the preset tunnel current is detected.

3.2.1 Topographic measurements.

The measurement control dialogue, as illustrated by figure 3.2.4 allows parameters used during topographic measurements to be tailored to the specific requirements of the material system under investigation. Parameters can be altered by use of a slider or by direct numeric entry. A brief summary of the most relevant features of the dialogue follows. The 'Gap voltage:' defines the bias applied between the tip and sample during scanning and can be set in the range +/- 0 to 10V. The 'Feedback Set:' defines the tunnel current which the feedback loop maintains during constant current imaging and can be selected in the range 0-5nA or 0-50nA, depending on the setting of the range button in the dialogue. The 'Loop Gain:' group allows adjustment of the feedback loop gain between 0.1 and 100%. This defines the response of the feedback loop and is tailored to suit the roughness of the material surface. The 'Scan Speed:' group defines the speed at which the tip is scanned across the sample, and is typically selected to be five times the magnitude of the frame size. For example, if the frame size is 200nm the scan speed would be 1000nm/s. The position and dimensions of the measured frame are displayed in 'Frame Position:' section of the dialogue, and can be tailored to suit requirements. During scanning the measured topographic information can be displayed as line profiles or colour graded images, examples of both are shown in figures 3.2.5 and 3.2.6 respectively. In the graded images, white areas are protrusions whereas black areas are depressions. The SCALA Pro software provides a suite of techniques to process information obtained during topographic measurements. These techniques are described fully within the SCALA software reference manual [2] and are not discussed here.

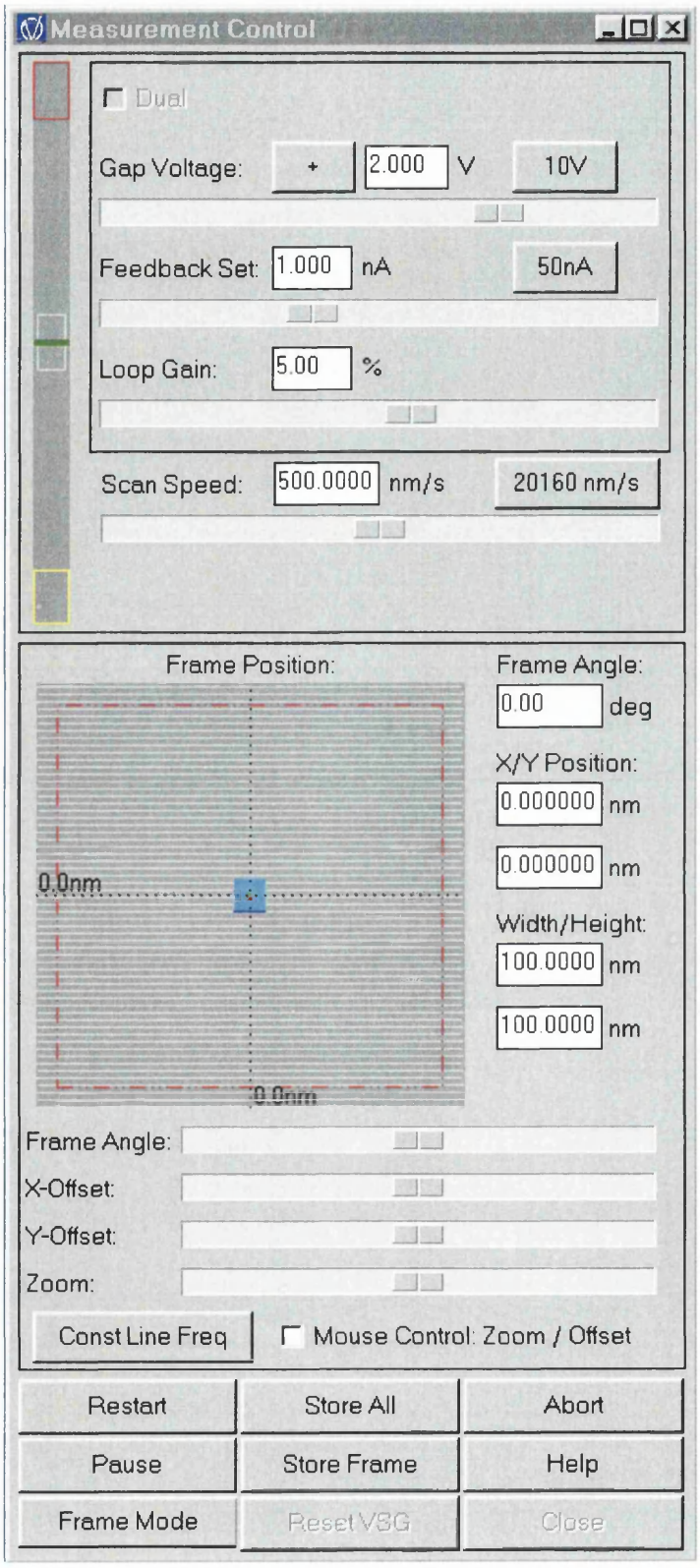


Figure 3.2.4: A screen shot illustrating the measurement control dialogue.

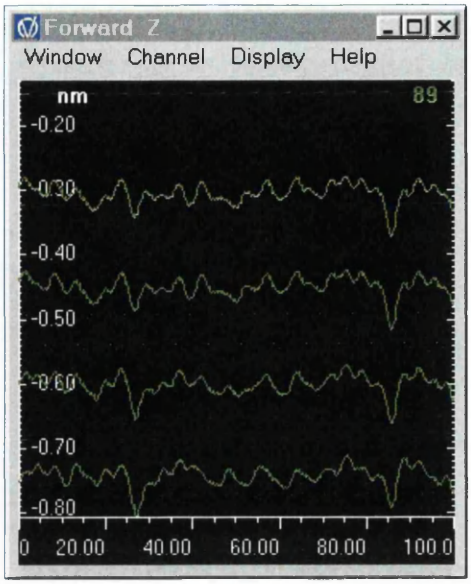


Figure 3.2.5: A screen shot of topographic data displayed as a line profile.

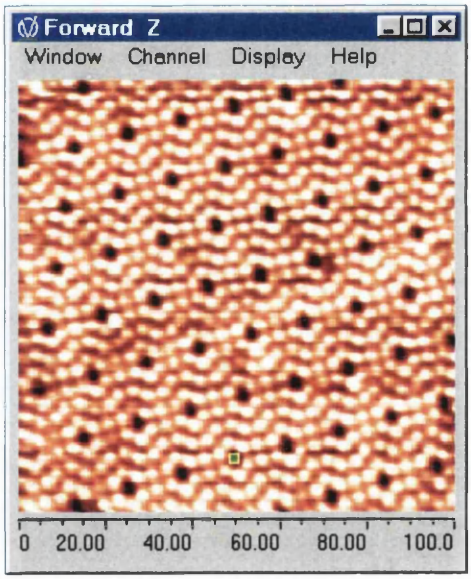


Figure 3.2.6: A screen shot of topographic data displayed as a colour graded image.

3.2.2 Spectroscopic measurements.

The spectroscopy preset dialogue, as illustrated by figure 3.2.7, allows parameters governing spectroscopy measurements to be tailored to suit the requirements of the material system being investigated. A brief discussion of the most salient features of the dialogue follows.

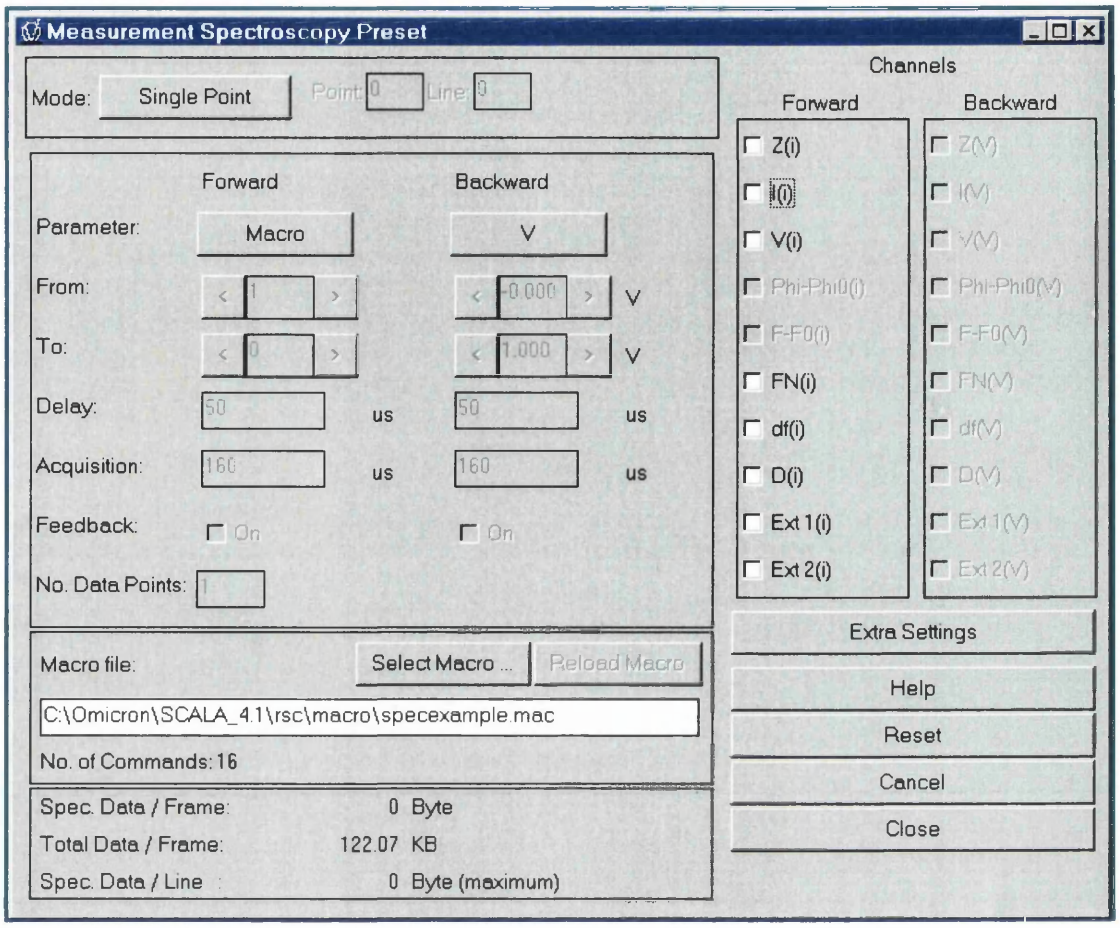


Figure 3.2.7: A screen shot illustrating the spectroscopy preset dialogue.

The 'Mode' button within the dialogue allows selection of the way in which spectroscopy is performed. Spectra may be acquired on a piecewise basis by manually selecting a point with the mouse, or alternatively on a regular basis at each point within the measurement frame or on a regular grid. Different parameter values can be selected for forward and backward scan lines; these are entered in the 'Forward' and 'Backward' sections respectively. The dependent variable (I, V, or Z) measured during spectroscopy is selected by tick boxes within the 'Channels' section of the dialogue.

At each measurement location, the SCALA interrupts the normal scan of the tip over the sample surface. When the tip is stationary the operation of the feedback loop is suspended to prevent the tip crashing into the surface as the bias voltage approaches 0V. The bias voltage applied between the tip and sample is ramped in discrete steps

from a nominal initial value to a nominal final value. Within the spectroscopy preset dialogue the 'From:' and 'To:' parameters define the two nominal values.

The number of discrete steps performed during the spectral acquisition is defined by the 'Data points' field. At each discrete step in bias voltage the tunnel current is measured for a period of time defined by the 'Acquisition:' field. The acquisition time defines the number of samples of the tunnel current performed at each discrete step in bias voltage. The acquisition time can take on values 2^n times the sample time, where the sample time is 10uS. The average of the tunnel current from all the samples forms the final value. Prior to measurement of the tunnel current at each discrete step in bias voltage, a delay is imposed to allow the system to settle, as defined within the 'Delay:' field.

The extra settings button produces a dialogue that allows additional parameters defining detailed aspects of the spectroscopy measurement to be configured, these mainly relate to the length of delays applied during spectral acquisition. Figure 3.2.8: Shows an oscilloscope trace of the sample bias voltage during acquisition of a single spectra.

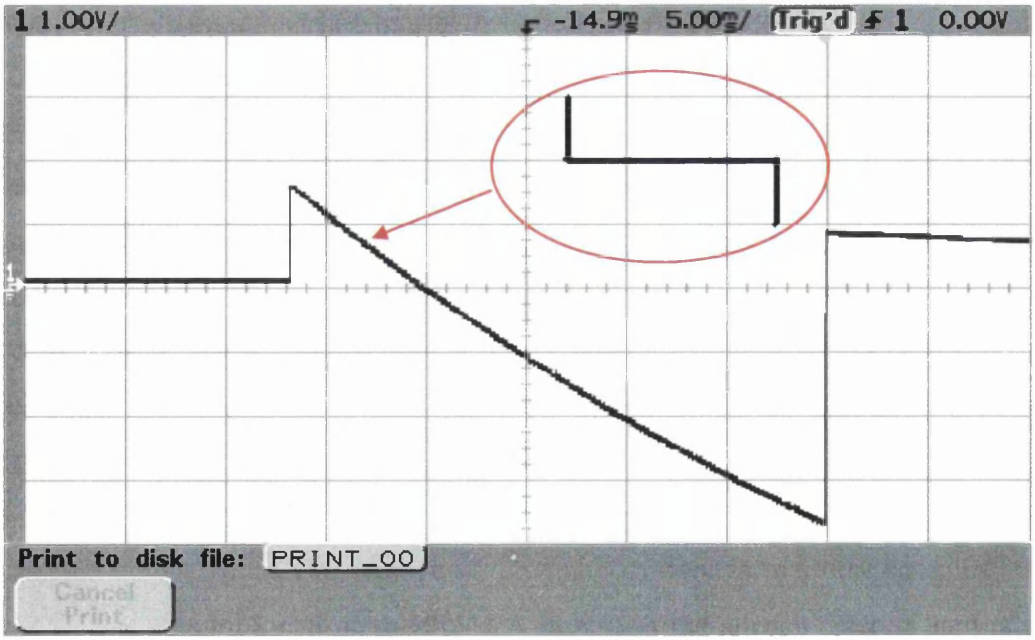


Figure 3.2.8: An Oscilloscope trace showing the bias voltage during acquisition of a single spectra containing 121 data points.

The spectra acquired during measurements is displayed in graphical format, as illustrated by figure 3.2.9.

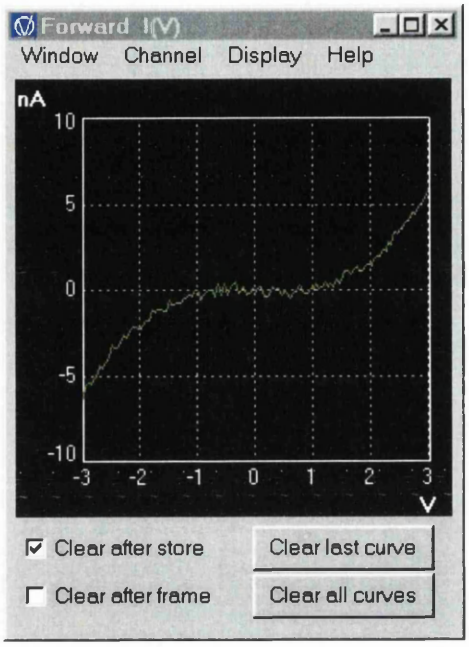


Figure 3.2.9: A screen shot illustrating the format in which information acquired during spectroscopic measurements is presented.

Macros used to perform customised spectroscopic measurements are selected from the spectroscopy preset dialogue. Choice of the 'Macro' option from the 'Parameter' list produced by toggling the parameter button, allows a macro file to be selected. The file is selected by toggling the 'Select Macro' button. The syntax of the macro file is checked automatically and errors displayed.

The SCALA Pro software provides a suite of techniques to process information obtained during spectroscopic measurements. These techniques are described fully within the SCALA Pro software reference manual [2] and are not discussed here.

3.2.3 The nano-structuring language.

The nano-structuring language provides facilities that can be employed to develop macro routines to perform both nano-structuring and complex spectroscopic measurements. Macros are composed within a suitable ASCII text editor using a command set of which a selection of pertinent commands is summarised by table

3.2.1. For further information regarding nano-structuring and spectroscopy macros refer to the Omicron SCALA Pro manual [2].

Command	Function
Feedback, flag	Controls the operation of the feedback loop. 'flag' = 0: feedback loop is disabled. 'flag' = 1: feedback loop is enabled.
Set_v, v	Sets the bias voltage to 'v', where v can be -10 to +10 v.
Add_v, dv	Increments the bias voltage by dV
Reset_v	Resets the bias voltage to the normal measurement value.
Set_z, z	Sets the tip height to 'z' nanometers with respect to the regulated height.
Add_z, dz	Increments 'z' by dV nanometers
Reset_z	Resets the tip height to the regulated value.
Set_signal, n	Sets status port signal 'n' (12,13,14,15) logically active, which corresponds to a TTL low output.
Reset_signal, n	Resets status port signal 'n' (12,13,14,15) inactive.
Delay, t	Delay for 't' microseconds, where t is 10, 20,...,32767
Loop_times, n	Repeat the following sequence of commends 'n' times.
Loop_end	Loop terminator
Reset_feedback	Resets the feedback loop status to the state that is selected for normal measurements.

Table 3.2.1: A summary of selected commands from the nano-structuring language.

3.2.4 Hardware interfacing to the SCALA.

External signal connections

A number of external connections are provided by the SCALA electronics to allow external monitoring of quantities involved in the measurement process or connection of instrumentation to be employed during the measurement process. A summary of the connections provided is given in table 3.2.2 along with a description of their function.

Connection	Function
Ext1,Ext2	External connections to the analog to digital converter. Input range -10 to +10V with 1M input resistance.
U EXT	External AC or DC voltage may be superimposed on the bias voltage. Input range -10 to +10V with 10K input resistance and 100KHz bandwidth.
Z OFFS	Allows an offset to be superimposed on the tip height. Input range -10 to +10V with 10K input resistance. 20mV is equivalent to 1Å offset.
I OUT	Tunnel current monitor output. An output of 10V = 50nA tunnel current.

Table 3.2.2 provides a summary of the connections provided by the SCALA along with descriptions of their functions.

Status port

The status port provides signals that enable monitoring / triggering of processes or events during operation of the STM, therefore enabling control of external instrumentation from the nano-structuring language. Physically the status port consists of a 25 way female D-type connector providing 16 output signals, 4 input channels and TTL (Transistor Transistor Logic) compatible power supplies. Of the 16 output channels 6 have predefined functions, 4 are accessible to the nano-structuring language and the remaining are reserved for future usage. All the output channels are active low open collector TTL compatible types. A summary of the signals provided by the status port is given by table 3.2.3 and the physical layout of the status port shown in figure 3.2.10.

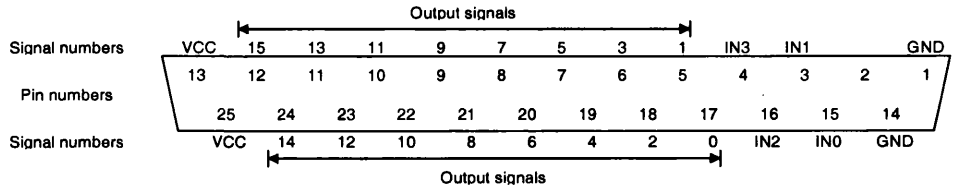


Figure 3.2.10: illustrates the signal number and associated physical pin number.

Pin number	Signal number	Signal name	Function
1		Gnd	Ground
2		None	None
3		IN1	Input channel 1 TTL logic compatible
4		IN3	Input channel 3 TTL logic compatible
5	1	SIG_L_FOR	Active while moving forward
6	3	SIG_SET_POINT	Inactive while averaging
7	5	Future use	None
8	7	SIG_FEEDBACK	Active when feedback loop is off
9	9	Future use	None
10	11	Future use	None
11	13	Recommended for Nano-Structuring	User defined from nano-structuring language
12	15	Recommended for Nano-Structuring	User defined from nano-structuring language
13		Vcc	+5 Vdc 100mA
14		Gnd	Ground
15		IN0	Input channel 0 TTL logic compatible
16		IN2	Input channel 2 TTL logic compatible
17	0	SIG_FRAME	Active while scanning
18	2	SIG_L_BACK	Active while moving backward
19	4	Future use	None
20	6	Future use	None
21	8	SIG_GAP	Active for 100uS when Vbias changes
22	10	Future use	None
23	12	Recommended for Nano-Structuring	User defined from nano-structuring language
24	14	Recommended for Nano-Structuring	User defined from nano-structuring language
25		Vcc	+5 Vdc 100mA

Table 3.2.3: A summary of the signals provided by the status port along with a description of their function.

3.3 Tip preparation.

Tips used during experimentation were prepared in house using a direct current electrochemical etching method as described by Ibe *et al* [4]. An electrochemical cell as shown in figure 3.3.1 is formed by immersing tungsten wire of 0.25mm diameter 2mm below the surface of a 2M saturated solution of KOH, which also forms the anode of the cell. The cathode of the cell is formed by the stainless steel beaker containing the solution.

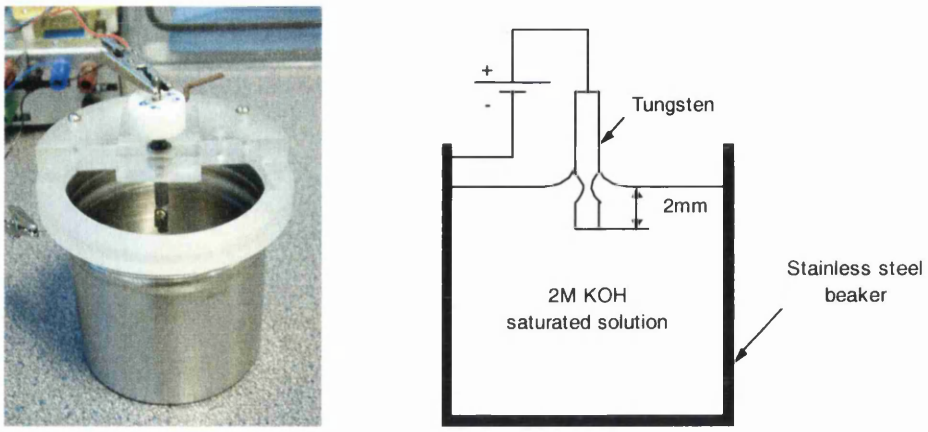


Figure 3.3.1: Schematic diagram and photograph of the electrochemical cell showing the tungsten wire (anode) and stainless steel beaker (cathode) containing the KOH solution.

Application of 2 -3 Volts DC between the anode and cathode causes oxidative dissolution of W to soluble tungstate anions at the anode and the reduction of water to form bubbles of hydrogen and and OH^- ions at the cathode. Etching occurs at the air solution interface and the reaction proceeds until the weight of the portion of the W wire below the surface exceeds the tensile strength of the etched region of wire, at which point the wire breaks. Just prior to the wire breaking the supply voltage is removed to ensure sharp tips are produced.

Etched tips are washed with deionised water to remove remnants of the KOH solution. Further cleaning to remove etching residuals such as oxides of W and hydrocarbons remaining on the tips surface is performed by field emission within the STM prior to usage.

3.4 References.

[1] S. Park, R.C. Barrett, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 2, **53 - 56**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[2] Omicron NanoTechnology, 'The SCALA Pro Software Manual', Version 4.1 (2001).

[3] Omicron NanoTechnology, 'SCALA Electronics Technical Reference Manual', Version 1.4 (2000).

[4] J.P. Ibe, P.P. Bey, S.L. Brandow, R.A. Brizzolara, N.A. Burnham, D.P. DiLella, K.P. Lee, C.R.K. Marrian and R.J. Colton, J. Vac. Sci. Technol. A **8**, 3570 (1990).

Chapter 4

Hardware Design

Introduction

Herein a description of the design and manufacture of hardware to implement variable tip-sample separation STS in conjunction with the Omicron SCALA electronics is discussed. Firstly, a brief description of the hardware functions necessary to implement variable tip-sample separation is discussed. A brief description of the method used by the SCALA electronics to implement fixed tip-sample separation STS is then given. Following this a general analysis of the requirements of variable tip-sample separation from the perspective of hardware is presented, and block level solution given and discussed. A detailed description of the design of each functional unit within the block level solution is presented, along with pertinent background information. Finally, a discussion of the design methodology and manufacturing process employed to produce printed circuit boards (PCBs) is presented.

4.1 The hardware requirements for Linear and Non-Linear variable tip-sample separation STS.

In order to implement both linear and non-linear variable tip-sample separation STS in conjunction with the Omicron SCALA, additional hardware is required to implement functions not provided by the SCALA electronics and SCALA pro software. The linear and non-linear tip-sample separation contours necessary in variable tip-sample separation measurements need to be implemented in external hardware. In addition, it is a requirement of the variable tip-sample separation technique to take conductivity measurements for every data point in an STS spectra. The SCALA electronics provide no facility to accomplish this; therefore external hardware is required to provide this function. Interfacing hardware is required to synchronise the operation of the SCALA electronics during STS measurements to the external hardware providing the aforementioned functions. Figure 4.1.1 show a block diagram of the proposed system, indicating the functionality provided by the Omicron SCALA and the external hardware.

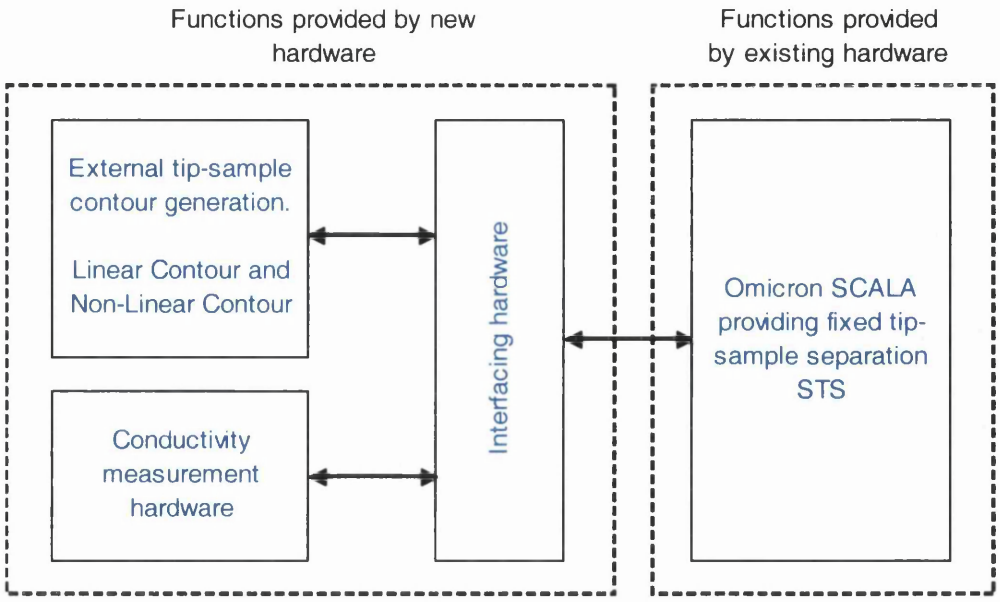
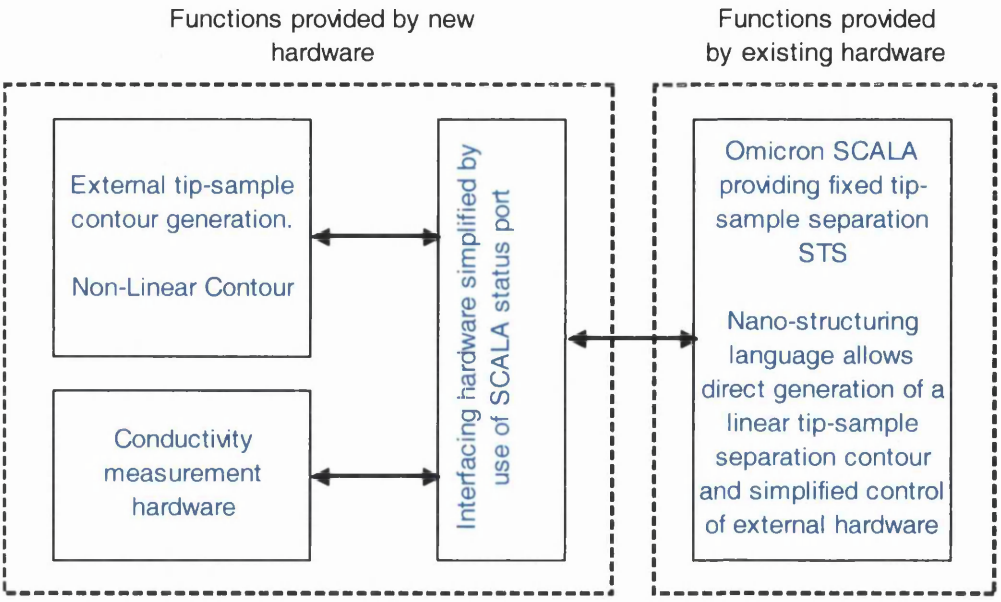


Figure 4.1.1: A block diagram showing the functions provided by the SCALA and external hardware.

During the initial stages of development a new version of the SCALA pro software was released that provided additional functionality within the nano-structuring language (details are given in chapter 3). The additional functions provided by the nano-structuring language allowed some of the functionality implemented exclusively within the external hardware to be duplicated by software routines. The nano-structuring language provides functions to facilitate control of a number of logic signals provided by the SCALA electronics status port. Therefore, allowing the interface between the SCALA electronics and the external hardware to be simplified. However, the restricted facilities provided by the nano-structuring language prevent development of software routines to generate non-linear tip-sample separation contours and carry out accurate conductivity measurements. These functions need to be implemented in external hardware. The modified block diagram of the proposed system, exploiting the reduction in the complexity of the external hardware afforded by the new software version is shown in figure 4.1.2.



4.2 Fixed tip-sample separation STS as implemented by the SCALA electronics.

In order to implement fixed tip-sample separation STS the SCALA electronics adopts a method similar to that described by Feenstra *et al* [1]. The operation of the feedback loop, which is required to perform constant current STM is interrupted. This is done to prevent feedback loop altering the tip-sample separation in order to maintain a constant tunnel current, and therefore prevent the tip crashing into the sample surface. After the feedback loop is disabled, the sample bias is ramped in discrete steps and the tunnel current measured for every discrete step in sample bias, as illustrated by figure 4.2.1. A number of delays are introduced into this acquisition process to allow the tip-sample system to stabilise prior to measurement of the tunnel current. Further details regarding the parameters which govern the measurement procedure can be located within the SCALA Pro software manual [2].

To implement variable tip-sample separation STS with the SCALA, it is necessary to create a tip-sample separation contour during the measurement, and to perform measurements of the sample conductivity for every data point within a spectra. In order to meet these requirements external instrumentation is required. In addition to the hardware required to perform the aforementioned tasks, a method of synchronising the operation of the SCALA electronics with the external instrumentation is necessary, to ensure correct chronology in the measurement.

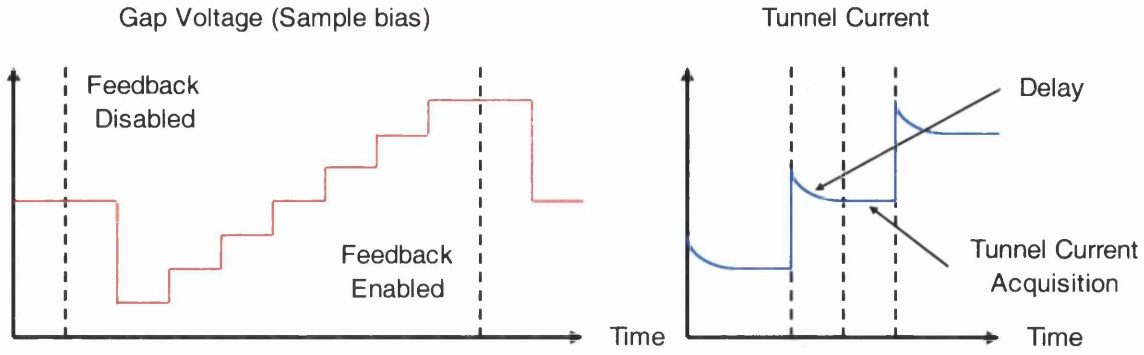


Figure 4.2.1 A schematic illustrating the operation of the SCALA electronics during fixed tip-sample separation STS.

4.3 Implementation of Variable tip-sample separation STS with the SCALA.

The hardware used to implement variable tip-sample separation STS in unison with the SCALA electronics, consists of two distinct subsystems that are responsible for conductivity measurement and generation of the tip-sample separation contour.

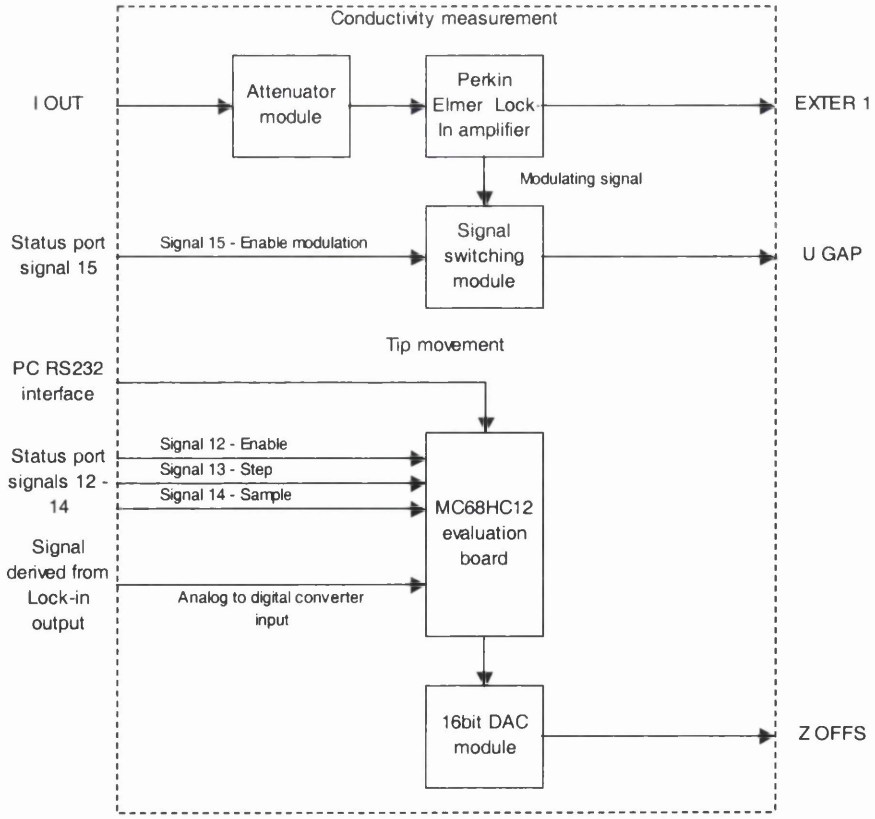


Figure 4.3.1: Functional block diagram of the implementing hardware.

Figure 4.3.1 shows the functional block diagram of the hardware with the two subsystems labelled as 'Conductivity measurement' and 'Tip movement'.

The hardware responsible for generating the tip-sample separation contour consists of a MC68HC912B32 microcontroller evaluation board and a 16bit digital to analog converter (DAC) module. The evaluation board provides a convenient platform on which solutions can be developed and rapidly implemented. The majority of support components required to operate the microcontroller as a stand-alone solution are provided, with the necessity for minimal additions. The 16bit DAC module converts the calculated binary offset produced by the microcontroller to an equivalent analog representation, which is applied to the SCALA electronics Z OFFS (Z offset) input during spectral acquisition.

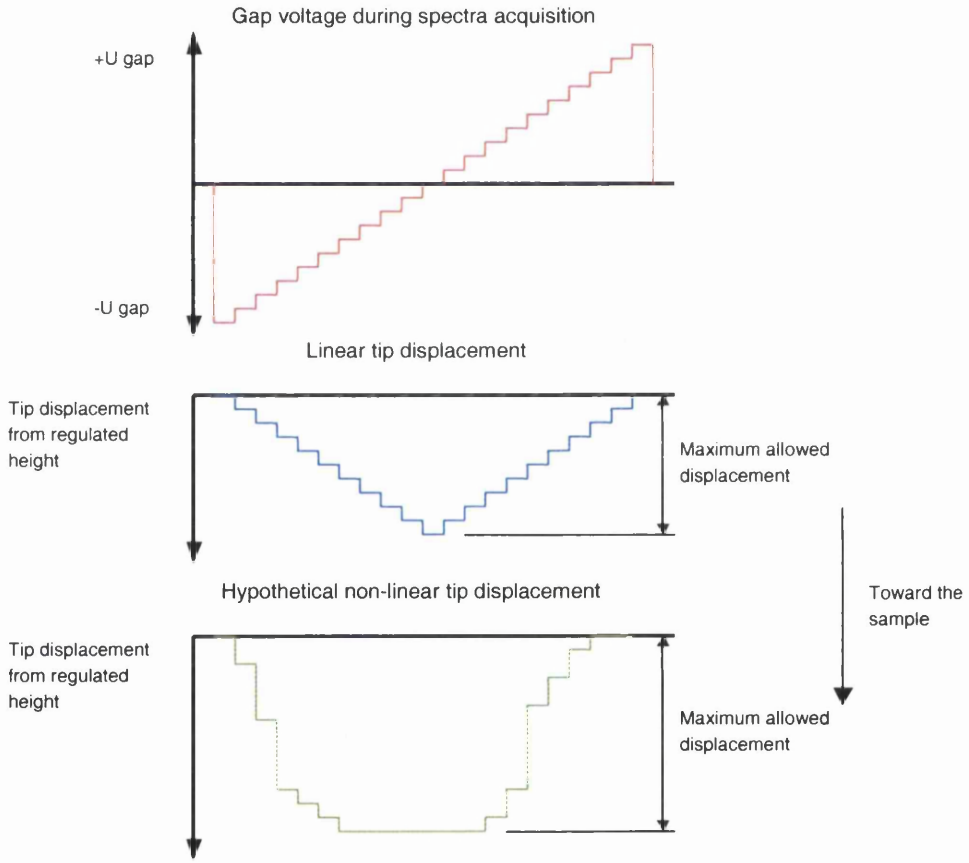


Figure 4.3.2: Linear and hypothetical non-linear tip-sample separation contours generated during variable tip-sample separation STS.

Presently both linear and non-linear programs are supported by the microcontroller, however development of additional hardware is required to complete the non-linear solution. Figure 4.3.2 illustrates schematically the tip-sample separation contours generated by the microcontroller during variable tip-sample separation STS.

For each discrete step in sample bias (Gap voltage) an offset is calculated and applied, creating an offset in tip displacement from the regulated height above the sample. In the case of the linear program, the offset is linearly related to the gap voltage and the slope, which is determined by the maximum allowable displacement. During the first half of acquisition of a spectra, as the gap voltage approaches zero from either maximum, the tip displacement is increased linearly with respect to the regulated height. At the point where the gap voltage crosses zero the direction of tip displacement is reversed and the offset reduced linearly. In the case of the non-linear program it is proposed that the offset be determined from the conductivity of the sample at each discrete step in gap voltage. Presently the non-linear program samples a voltage presented to the microcontroller's analog to digital converter (ADC) and determines an appropriate offset. Details regarding both linear and non-linear programs can be located in chapter 5.

Signal 12 interfaces to the microcontroller and serves to indicate the initiation and completion of acquisition of a complete spectra. Signals 13 and 14 order events during acquisition of the spectra; signal 14 being exclusively used with the non-linear offsets. On a falling edge transition from logically inactive to active states signals 13 and 14 both cause interrupts within the microcontroller. Following an interrupt from signal 13 the microcontroller calculates and applies an offset voltage to the SCALA Z OFFS input, causing an offset from the regulated tip height proportional to the applied voltage. An interrupt caused by signal 14 initiates sampling of the analog voltage presented to the microcontrollers 8bit ADC. The result of the sampling is used in determination of the offset voltage for the non-linear program. Figure 4.3.3 shows the timing diagram for status port signals 12,13 and 14 for both linear and non-linear programs.

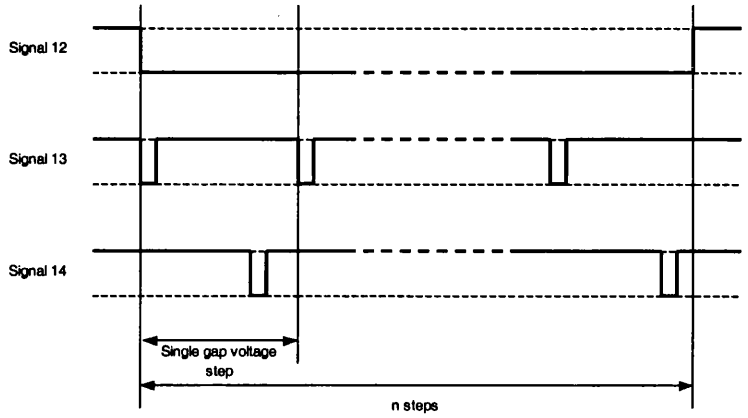


Figure 4.2.3: Timing diagram of status port signals 12,13 and 14 during acquisition of a single spectra.

Serial communication between the microcontroller and SCALA host PC is provided to facilitate transfer of program parameters from the PC to the microcontroller. The microcontrollers serial communications interface (SCI) is utilised to implement a standard RS232 format interface, which connects to the SCALA electronics host PC serial communications (COMM) port.

The hardware used to perform conductivity measurements consists of a Lock-in amplifier, Signal Routing module and Attenuator. During each discrete step in the gap voltage, conductivity measurements are performed by modulating the gap voltage for a period of time with a sinusoidal reference signal of known frequency and amplitude and measuring the resultant modulation in tunnel current. A Perkin Elmer Lock-in amplifier both generates the reference signal and deduces the conductivity from the tunnel current and reference signals, producing an output voltage proportional to the conductivity that is logged using one of the SCALA electronics external ADC inputs (EXTER 1). Analog switches within the switching module route the reference signal from the lock-in to the SCALA external gap voltage input (U GAP) under control of the status port. Status port signal 15 is used to control the switching module. The lock-in amplifier input is derived from the SCALA tunnel current monitor output (I OUT) and attenuated if necessary; the degree of attenuation depending on the conductivity of the sample under investigation. The measurement process is schematically illustrated by figure 4.3.4.

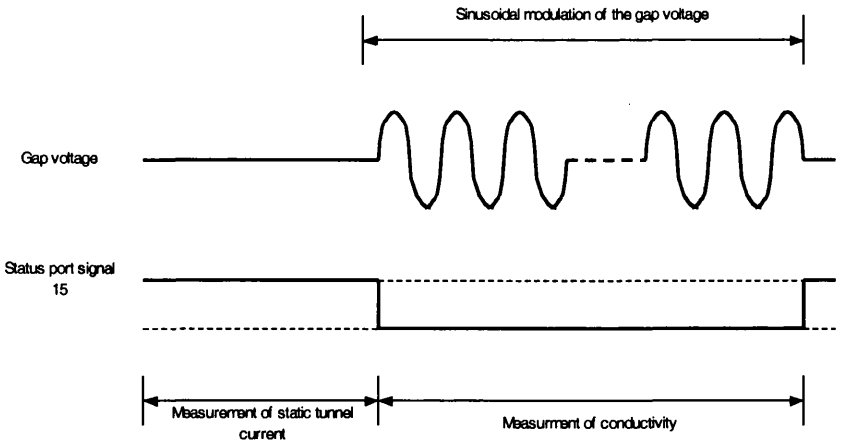


Figure 4.3.4: Conductivity measurement for a single step in gap voltage.

4.4 The MC68HC912B32 microcontroller and P&E Microsystems evaluation board.

The MC68HC12 microcontroller is described briefly in section 4.4.1 with emphasis placed on features utilised by the present application. Further to this, the P&E microcomputer systems evaluation board around which the system has been developed and modifications to that board are then discussed in section 4.4.2.

4.4.1 The MC68HC12 microcontroller.

The MC68HC12 family of microcontrollers incorporate a common 16bit central processing unit, designated CPU12 and a number of common on chip peripheral devices. Figure 4.4.1 shows the simplified functional block diagram of the MC68HC912B32 as found on the evaluation board. Details regarding peripheral devices utilised in the present application are discussed in the following paragraphs with reference to Figure 4.4.1. For a more comprehensive treatment of any of the following topics, the reader is referred to Motorola reference manuals for the microcontroller [3], [4]. The microcontroller can be operated in a number of modes that determine configuration of the internal memory and port arrangement, these are: single chip, expanded narrow and expanded wide. Selection of operating mode is made via the logic levels applied to mode pins MODA and MODB and the background debug pin BKDB, during reset.

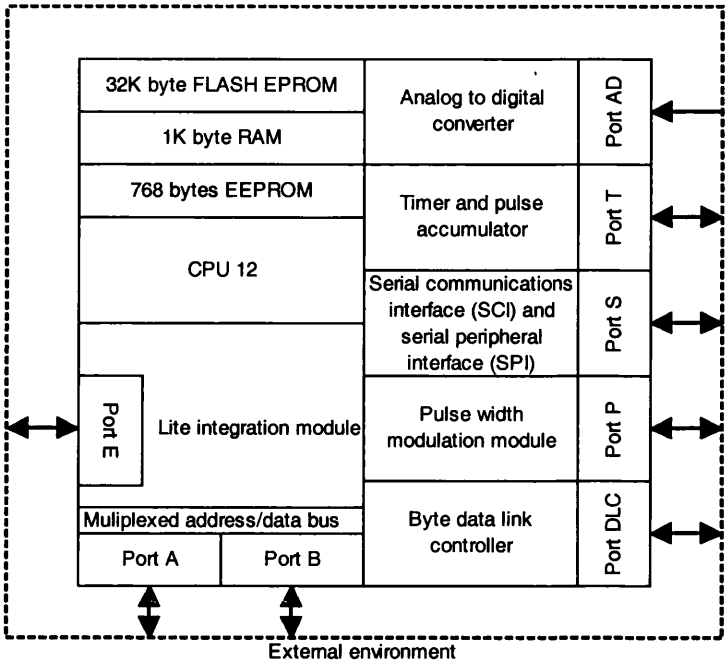


Figure 4.4.1: Simplified functional block diagram for the MC68HC912B32.

Currently the microcontroller is operated in single chip mode in which there are no external busses; therefore all ports are available for interfacing purposes. In addition to standard operating modes, background debug mode is used during system development. This is an auxiliary operating mode that provides debug options that aid program debug during system development. Code may be executed normally whilst register and memory locations are interrogated, thus facilitating real time monitoring of system resources. Memory allocation for single chip operation is schematically illustrated by figure 4.4.2. Three types of memory are contained within the microcontroller: 32K bytes of Flash EPROM, 768 bytes of EEPROM and 1K byte of RAM. The 32K byte block of flash is mapped into the upper half of the addressable memory space, address range $8000-$FFFF. In the default single chip mode, the RAM and EEPROM are mapped to address ranges $0800-$0BFF and $0D00-$0FFF respectively, however these address ranges are not fixed and may be reconfigured. Presently the default memory configuration is employed.

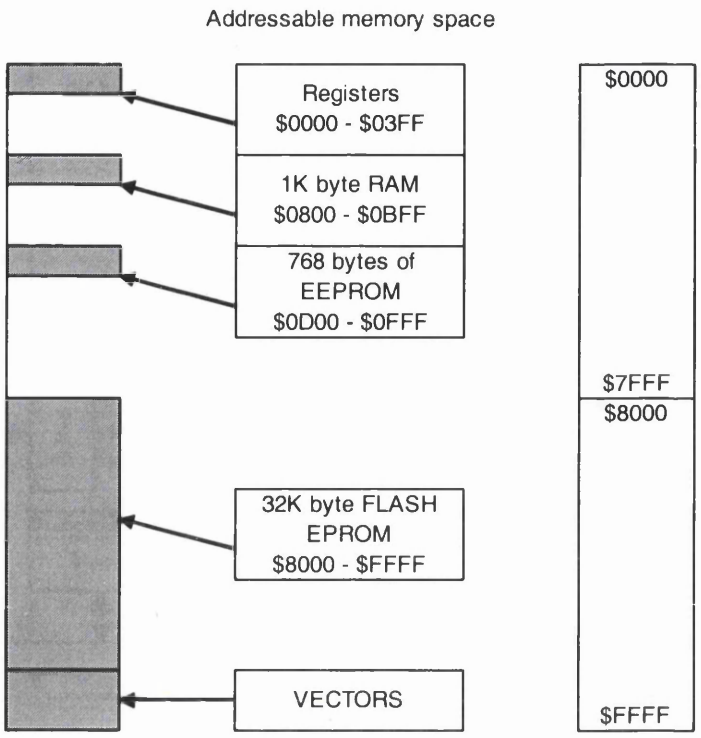


Figure 4.4.2: Memory map for default single chip mode.

The microcontroller incorporates asynchronous and synchronous serial interfaces; serial communications interface (SCI) and serial peripheral interface (SPI) respectively. Presently the SCI is used to implement an RS232 compatible communications link with the SCALA electronics host PC.

The microcontroller incorporates a 10bit successive approximation type ADC with 8 input channels that is configurable for one of eight modes of operation. Currently the converter is configured for 8 bit resolution and to perform a single scan on each input channel. An analog voltage in the range 0 to 5v presented to PORTAD 0 – 7 is converted to its equivalent binary representation.

The microcontroller incorporates a multi purpose standard timer module, which can be used for many purposes including measuring input waveforms and generating precise pulses. Eight configurable input capture/output compare channels are included in the timer module, each having a maskable interrupt associated with it. In the present application timer channels 0 and 1 are configured to generate hardware interrupts on the falling edge of an input waveform.

4.4.2 The P&E Microsystems evaluation board.

Figure 4.4.3 shows a picture of the P&E Microsystems MC68HC12 evaluation board that forms the development platform around which the variable tip-sample separation instrumentation is based. The board incorporates the bulk of necessary support circuitry to operate the microcontroller in normal and background single chip modes; however additional hardware is required to operate in expanded modes. A background debug interface is incorporated, through which the microcontroller can be programmed and code evaluated and debugged.

To enable selection of an operating mode the board has been modified by addition of the circuit shown in figure 4.4.4. SW1 to SW3 and R1 to R3 facilitate selection of logic voltages presented to MODA, MODB and BKDB pins and therefore the operating mode. Further information regarding P&E Microsystems development tools can found from their web site [5].

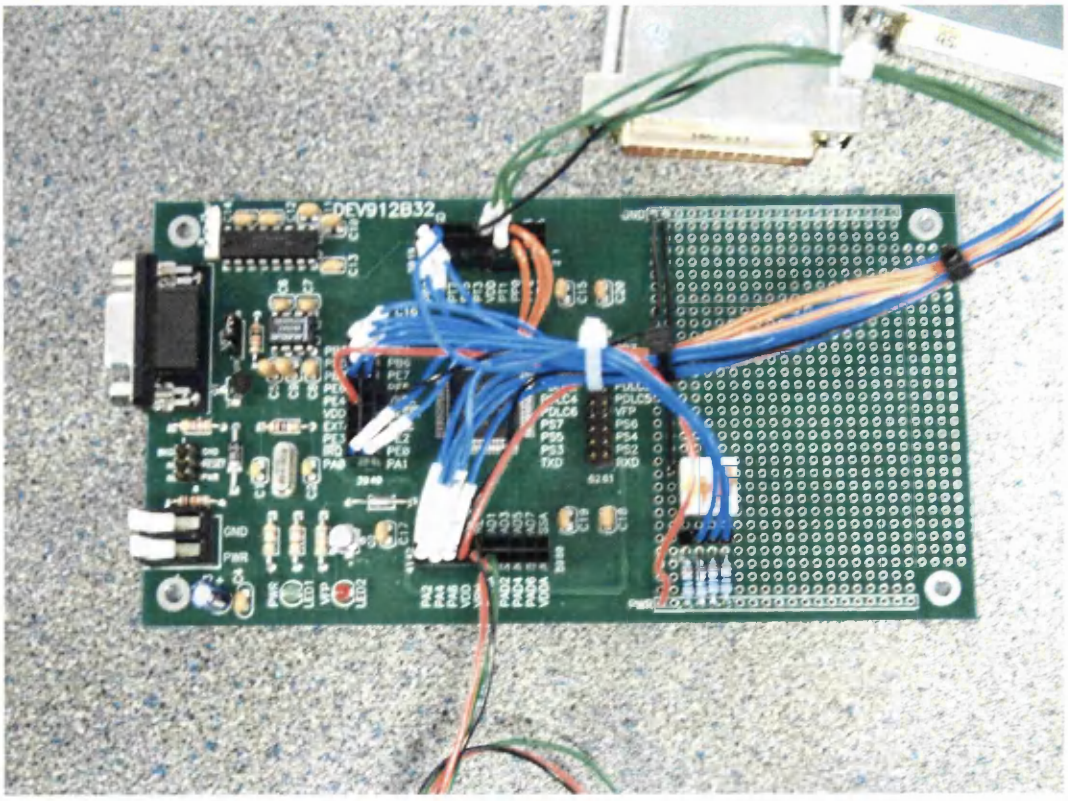


Figure 4.4.3: A picture of the P&E Microsystems evaluation board.

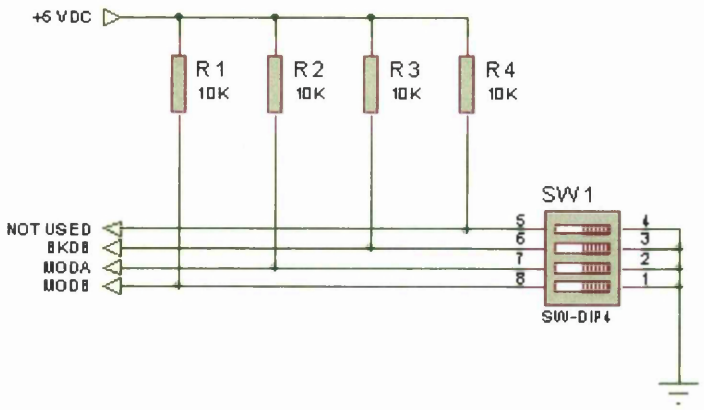


Figure 4.4.4: Circuit diagram of mode selection circuitry.

4.5 The Digital to Analog converter (DAC) module.

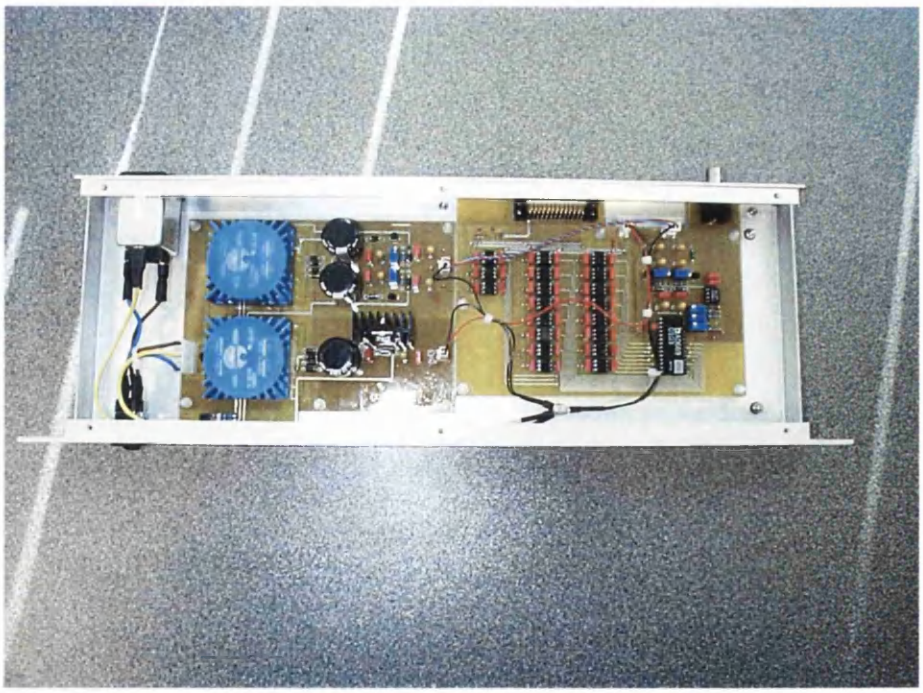


Figure 4.5.1: A picture of the DAC module and power supply encased in a 1U rack case.

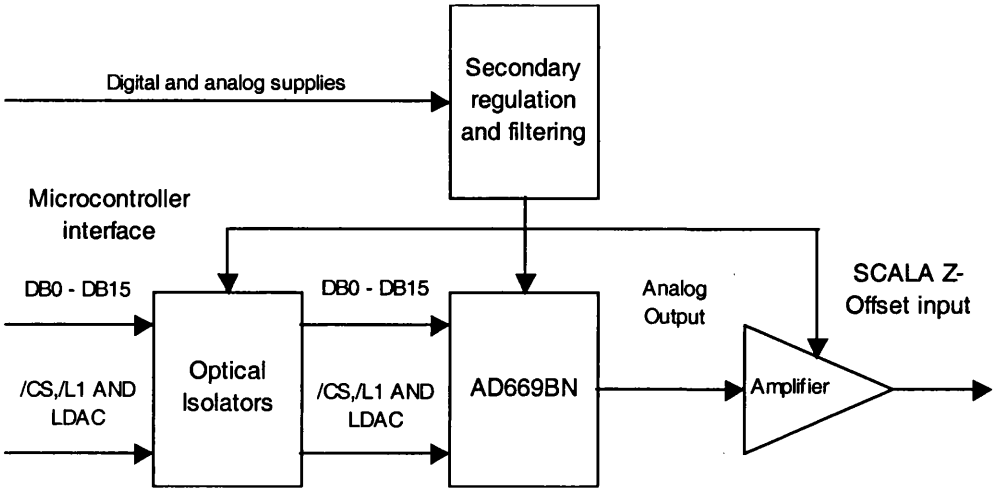


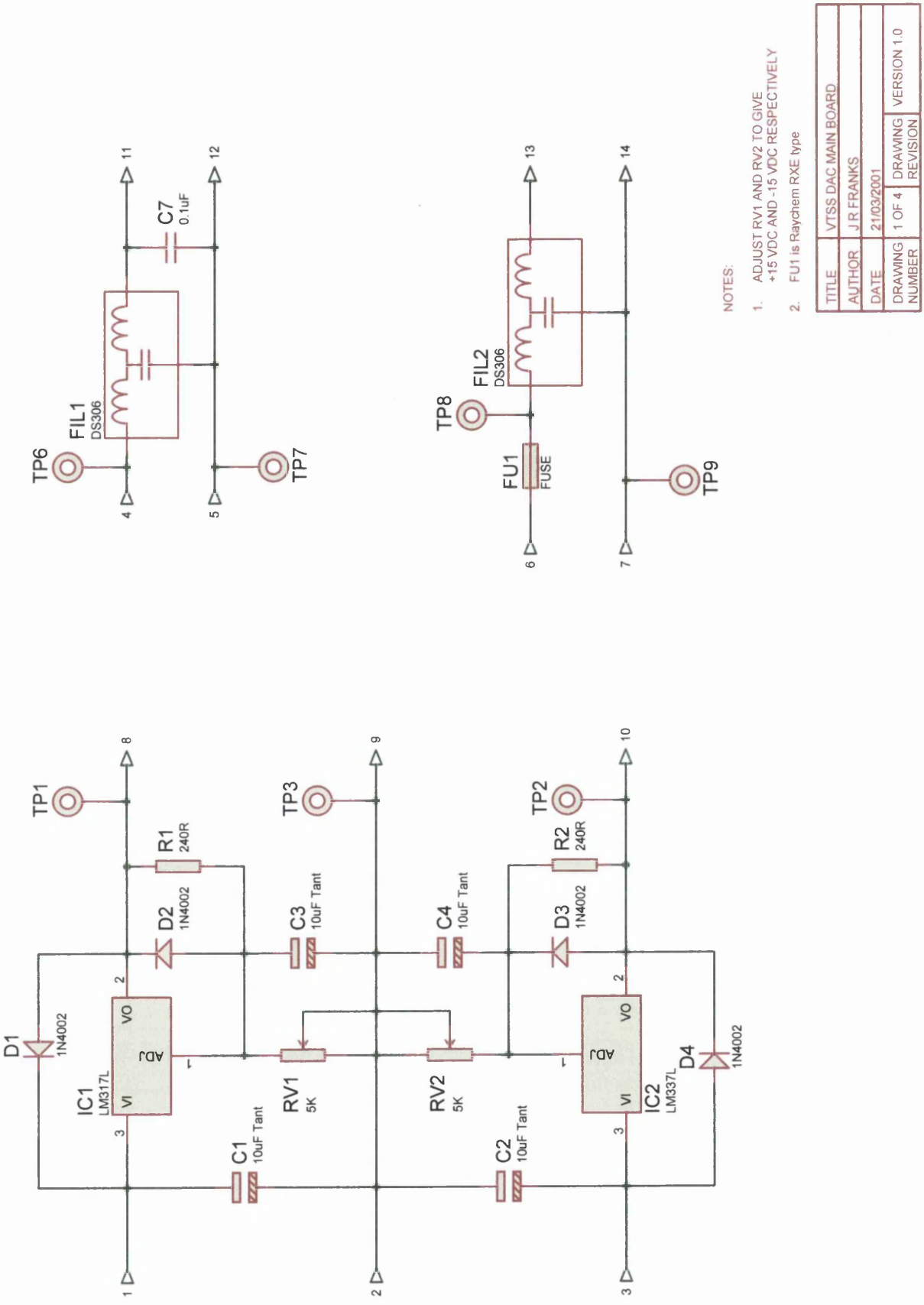
Figure 4.5.2: Functional block diagram of the DAC circuitry.

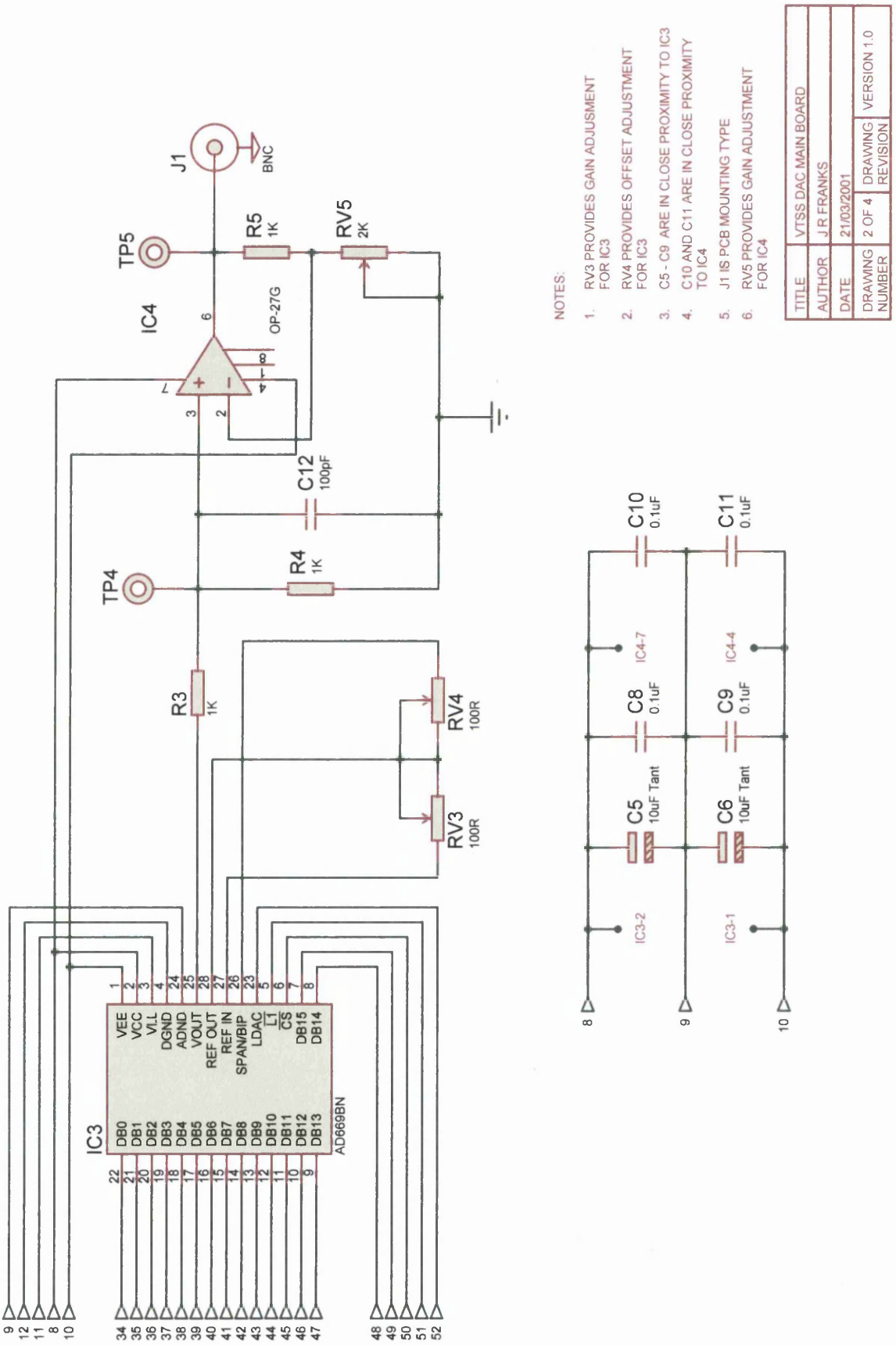
Fig 4.5.1 shows a photo of the assembled DAC module complete with power supply encased in 1U rack case. Fig 4.5.2 shows the functional block diagram of the complete digital to analog (DAC) converter module. The DAC module is based around an Analog Devices AD669 16-bit DAC, configured to provide a bipolar output. The output from the converter is attenuated by a factor of 50, which equates to +/- 100nm potential tip movement in Z. Opto-isolators are included in the digital signal paths to isolate the microcontroller from the converter, preventing contamination of the converter output with noise. Secondary regulation of the +/- 20V_{DC} supply ensures a clean supply to the converter and output scaling circuitry.

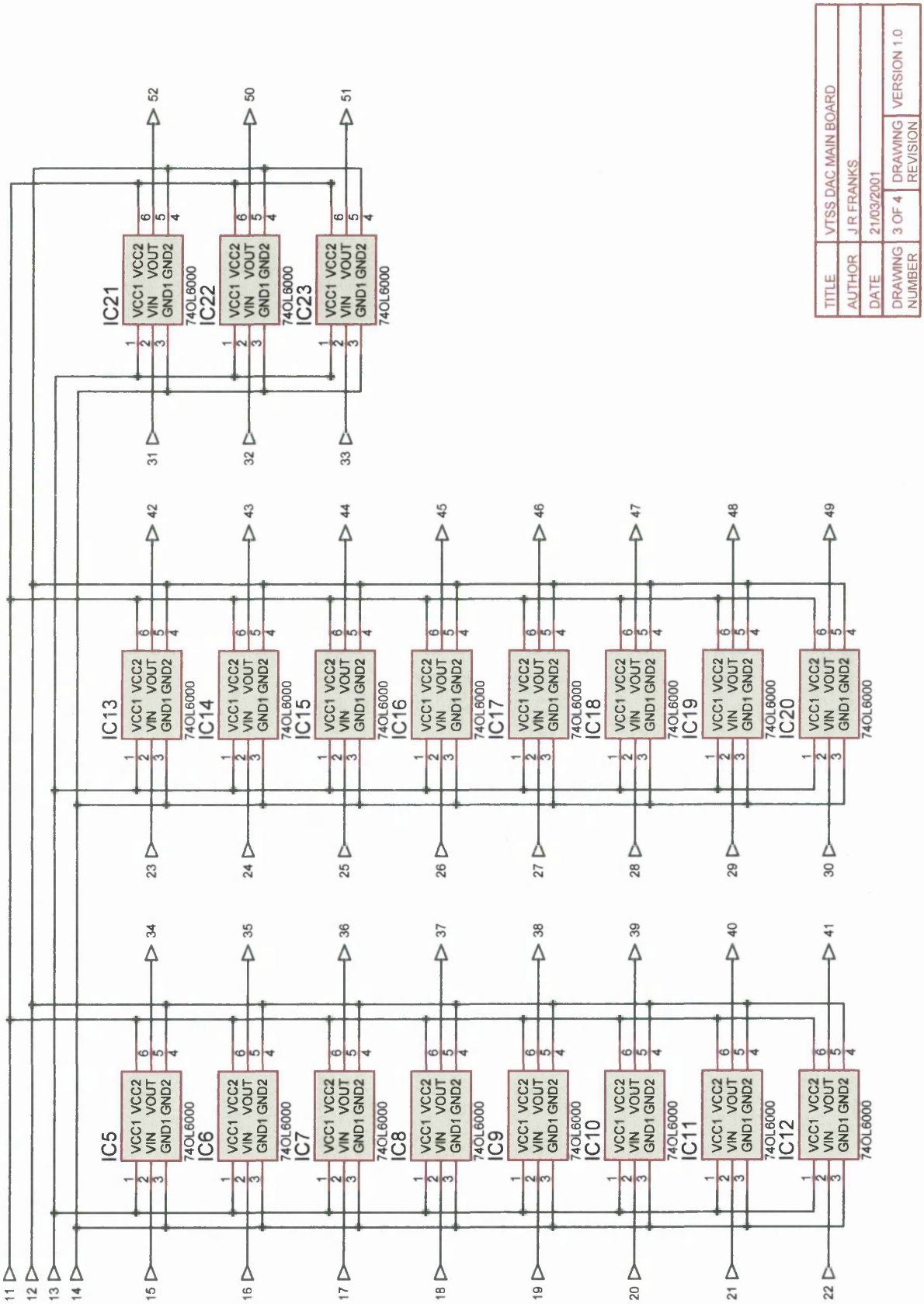
A detailed description of each functional unit in Fig 4.5.2 is presented in section 4.5.2, with reference to the circuit schematic for the DAC module, Fig 4.5.3. Pertinent background information regarding the AD669 is presented in section 4.5.1.

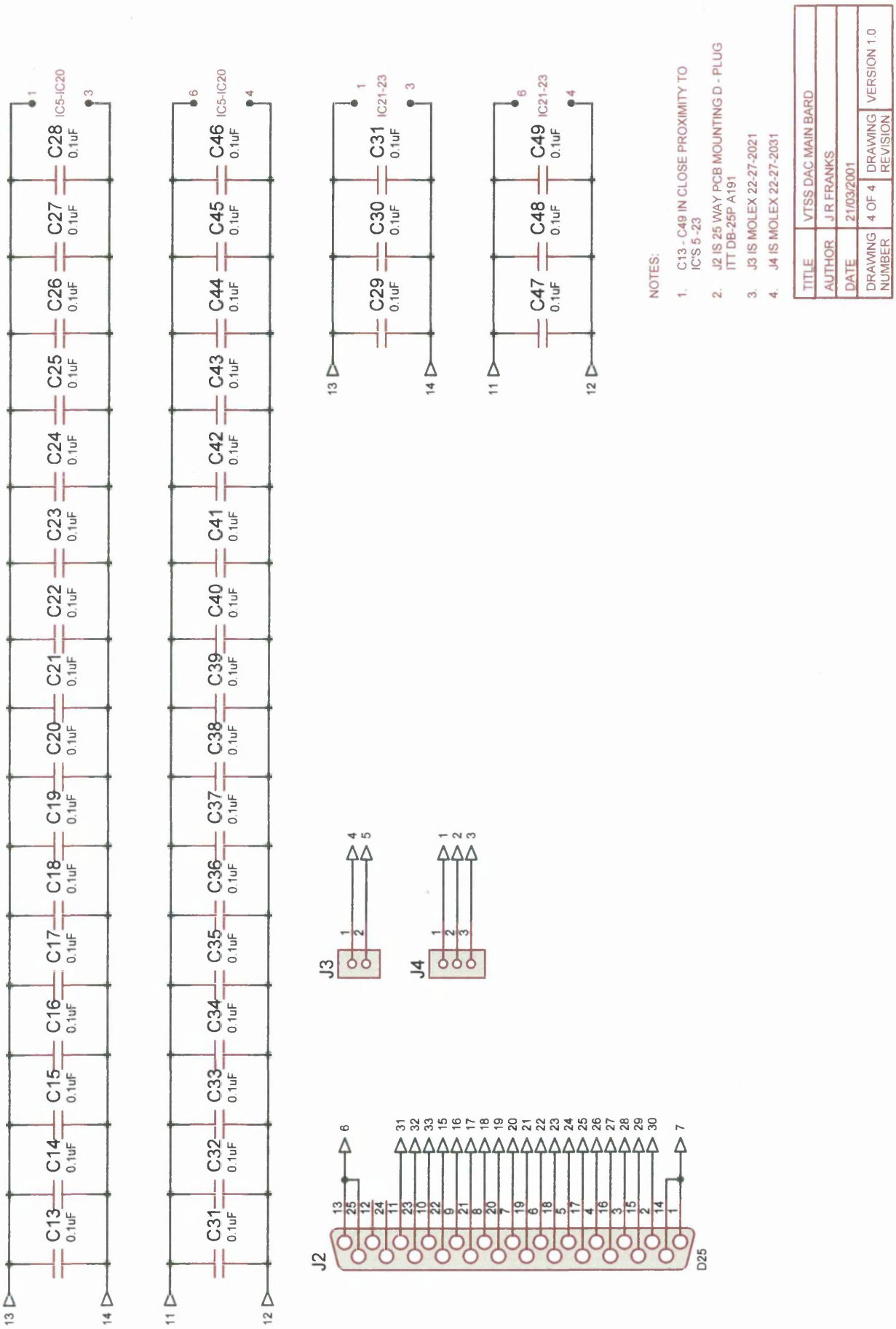
4.5.1 The AD669 16-bit DAC.

The AD669 [6] is a monolithic 16-bit digital to analog converter (DAC) with an on-board voltage reference and output amplifier. A simplified functional block diagram for the converter is shown in Fig 4.5.4.


TP6
FIL1
DS306
C7
0.1uF
TP7
TP8
FIL2
DS306
FU1
FUSE
TP9
NOTES:
1. ADJUST RV1 AND RV2 TO GIVE +15 VDC AND -15 VDC RESPECTIVELY
2. FU1 is Raychem RXE type
TITLE VTSS DAC MAIN BOARD
AUTHOR J R FRANKS
DATE 21/03/2001
DRAWING NUMBER 1 OF 4
DRAWING REVISION VERSION 1.0
TP1
R1
240R
D2
1N4002
TP3
C3
10uF Tant
C4
10uF Tant
TP2
R2
240R
D3
1N4002
D1
1N4002
IC1
LM317L
VI
VO
ADJ
C1
10uF Tant
RV1
5K
RV2
5K
C2
10uF Tant
IC2
LM337L
D4
1N4002


J1
BNC
TP5
R5
1K
RV5
2K
IC4
OP-27G
C12
100pF
TP4
R4
1K
R3
1K
RV4
100R
RV3
100R
IC3
AD669BN
VEE
VCC
VLL
DGND
ADND
VOUT
REF OUT
REF IN
SPAN/BIP
LDAC
CS
DB15
DB14
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
DB8
DB9
DB10
DB11
DB12
DB13
C10
0.1uF
C11
0.1uF
C8
0.1uF
C9
0.1uF
C5
10uF Tant
C6
10uF Tant
IC4-7
IC4-4
IC3-2
IC3-1
NOTES:
1. RV3 PROVIDES GAIN ADJUSMENT FOR IC3
2. RV4 PROVIDES OFFSET ADJUSTMENT FOR IC3
3. C5 - C9 ARE IN CLOSE PROXIMITY TO IC3
4. C10 AND C11 ARE IN CLOSE PROXIMITY TO IC4
5. J1 IS PCB MOUNTING TYPE
6. RV5 PROVIDES GAIN ADJUSTMENT FOR IC4
TITLE
VTSS DAC MAIN BOARD
AUTHOR
J R FRANKS
DATE
21/03/2001
DRAWING NUMBER
2 OF 4
DRAWING REVISION
VERSION 1.0


TITLE VTSS DAC MAIN BOARD
AUTHOR J R FRANKS
DATE 21/03/2001
DRAWING NUMBER 3 OF 4
DRAWING REVISION VERSION 1.0
IC5 IC6 IC7 IC8 IC9 IC10 IC11 IC12 IC13 IC14 IC15 IC16 IC17 IC18 IC19 IC20 IC21 IC22 IC23
74OL6000
VCC1 VCC2 VIN VOUT GND1 GND2


C13 0.1uF
C14 0.1uF
C15 0.1uF
C16 0.1uF
C17 0.1uF
C18 0.1uF
C19 0.1uF
C20 0.1uF
C21 0.1uF
C22 0.1uF
C23 0.1uF
C24 0.1uF
C25 0.1uF
C26 0.1uF
C27 0.1uF
C28 0.1uF
IC5-IC20
C31 0.1uF
C32 0.1uF
C33 0.1uF
C34 0.1uF
C35 0.1uF
C36 0.1uF
C37 0.1uF
C38 0.1uF
C39 0.1uF
C40 0.1uF
C41 0.1uF
C42 0.1uF
C43 0.1uF
C44 0.1uF
C45 0.1uF
C46 0.1uF
IC5-IC20
C29 0.1uF
C30 0.1uF
C31 0.1uF
IC21-23
C47 0.1uF
C48 0.1uF
C49 0.1uF
IC21-23
J2
D25
J3
J4
NOTES:
1. C13 - C49 IN CLOSE PROXIMITY TO IC'S 5 -23
2. J2 IS 25 WAY PCB MOUNTING D - PLUG ITT DB-25P A191
3. J3 IS MOLEX 22-27-2021
4. J4 IS MOLEX 22-27-2031
TITLE VTSS DAC MAIN BARD
AUTHOR J R FRANKS
DATE 21/03/2001
DRAWING NUMBER 4 OF 4
DRAWING REVISION VERSION 1.0

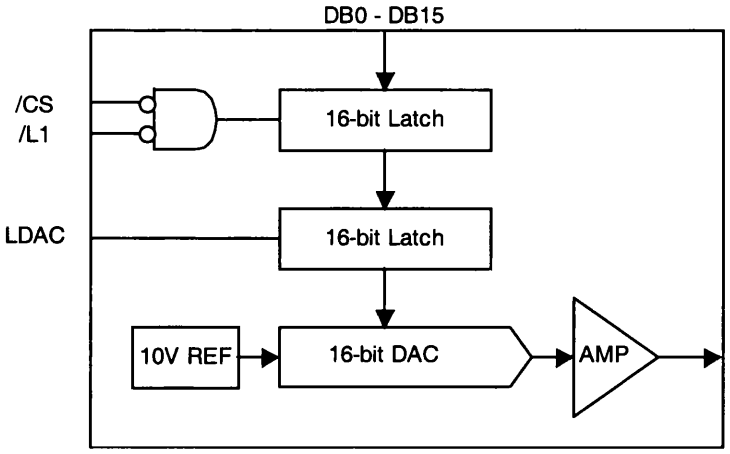


Figure 4.5.4: Functional block diagram of the AD669.

Data is loaded into the converter in 16-bit parallel format (DB0 – DB15). Two banks of latches ensure elimination of data skew errors and provide double buffering of the data. The first bank of latches is controlled via active low chip-select (/CS) and load (/L1) signals. The second bank is controlled via a single active high load-DAC (LDAC) signal. Fig 4.5.5 shows a simplified timing diagram of /CS, /L1 and LDAC to load the converter with a data word. Data is latched into the first bank of latches during the time /CS and /L1 are low and then transferred to the second bank during the time LDAC is high.

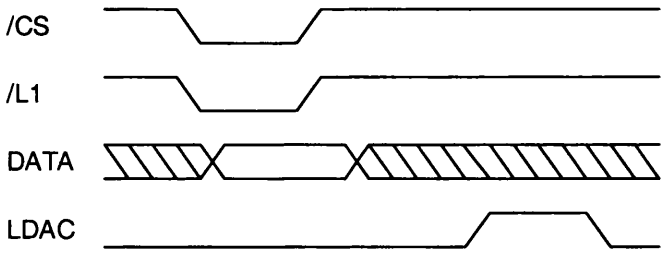


Figure 4.5.5: Timing diagram of a typical write cycle.

The converter may be configured for either unipolar (0 to +10V output) or bipolar (-10 to +10V output) operation. The resolution for each is 16-bits over the full output range, which yields 1 LSB as 153μV for unipolar and 305μV for bipolar operation. The converter uses positive-true binary coding, where a data word 0000H represents

0V and minus full scale in unipolar and bipolar configurations respectively, and FFFFH represents 1 LSB less than positive full scale in both cases.

The converter has separate analog (-VEE and VCC) and digital (VLL) power supplies with separate grounds to minimise coupling of noise from the digital section of the converter to the analog section.

4.5.2 Description of the DAC module circuitry.

In the circuit schematic shown in figure 4.5.3, IC3 is connected in bipolar configuration with RV3 and RV4 providing gain adjustment for the converters internal amplifier and offset adjustment respectively. Each of the 16 data lines (DB0 – DB15) and 3 control lines (LDAC, /L1 and /CS) of IC3 are optically isolated from the microcontroller ports with IC5 – IC23 [7]. These devices are opto-isolators produced by Quality Technologies and incorporate TTL compatible interfaces, making them ideal for the present application. The maximum operating frequency of the devices is 15Mhz, which is ample in the current application. Each opto-isolator requires the addition of two 100nF decoupling capacitors across the transmitter and receiver power supplies; C13 to C49 fulfil this function.

The +/- 10V bipolar output from IC3 is attenuated and low pass filtered by R3, R4 and C12. The values of R3 and R4 were selected to attenuate the output of IC3 by a factor of approximately 100, which corresponds to an output voltage range of +/- 100mV. The combination of R4 and C12 produces a low pass filter with a –3dB point at 16 KHz. The effect of the filter is to remove higher frequency components from the output of IC3, therefore limiting the slew rate of the output. Additionally this reduces the likelihood of ringing and stability problems.

The output from the attenuator and low pass filter is amplified by IC4, which is arranged as an adjustable non-inverting amplifier. The gain of IC4 is determined by R5 and RV5 and is set nominally to two, a value that corresponds to a +/- 200mV output voltage range or +/- 100nm tip movement range. In practice the gain of IC4 is adjusted to compensate for scaling inaccuracies within the SCALA electronics.

Linear adjustable voltage regulators IC1 and IC2 (LM317LZ and LM337LZ respectively) reduce the +/- 20VDC supply from the power supply board to +/- 15VDC, as required by IC3 and IC4. The output voltage of the regulators is determined by R1 and RV1 for IC1, and R2 and RV2 for IC2. Secondary regulation as performed by IC1 and IC2 ensures the supply to IC3 and IC4 is as clean as possible. A detailed description of the use of IC1 and IC2 is discussed in section 4.7 and therefore not reproduced here. The supplies to the transmitters and receivers of IC5 – IC23 are filtered using Murata three terminal filters, FIL1 and FIL2. The function of these filters is to attenuate high frequency noise on both 5VDC supplies.

Care has been taken in the PCB layout to ensure that decoupling capacitors are physically positioned in close proximity to their respective IC's. IC1 and IC2 are located in close proximity to IC3, and interconnections between them kept as short as possible, thus reducing the likelihood of introducing noise into the supplies of IC3. Extensive use of ground planes has also been adopted for similar reasons. PCB artwork for the DAC module is located in appendix A.

4.6 The Signal Routing module.

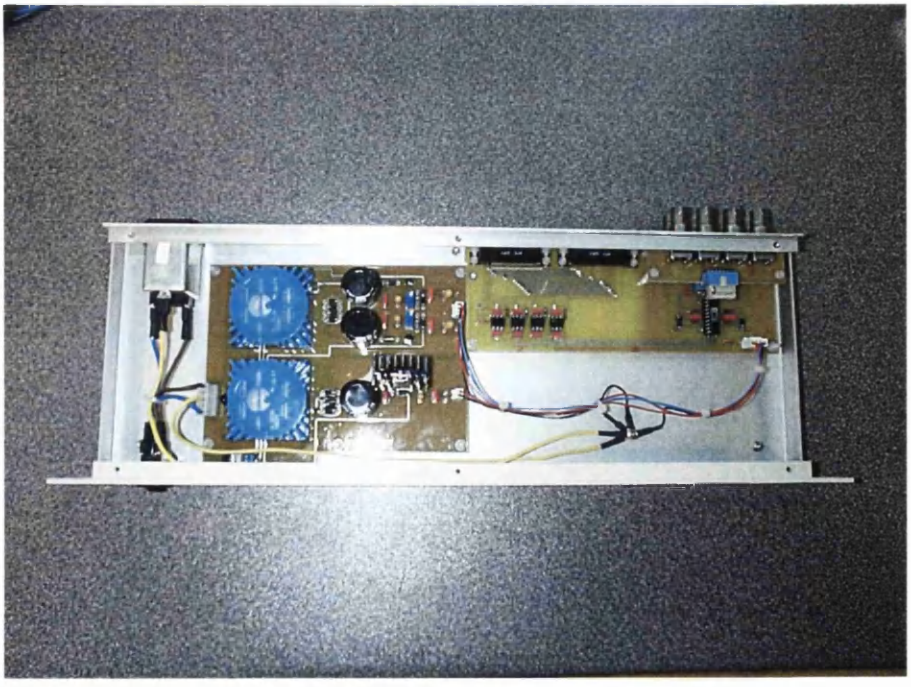


Figure 4.6.1: A picture of the Signal Routing module and power supply encased in a 1U rack case.

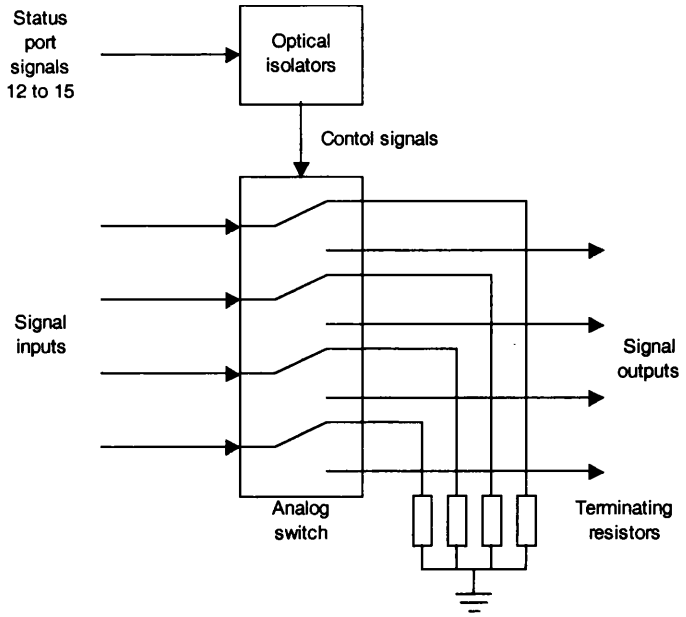
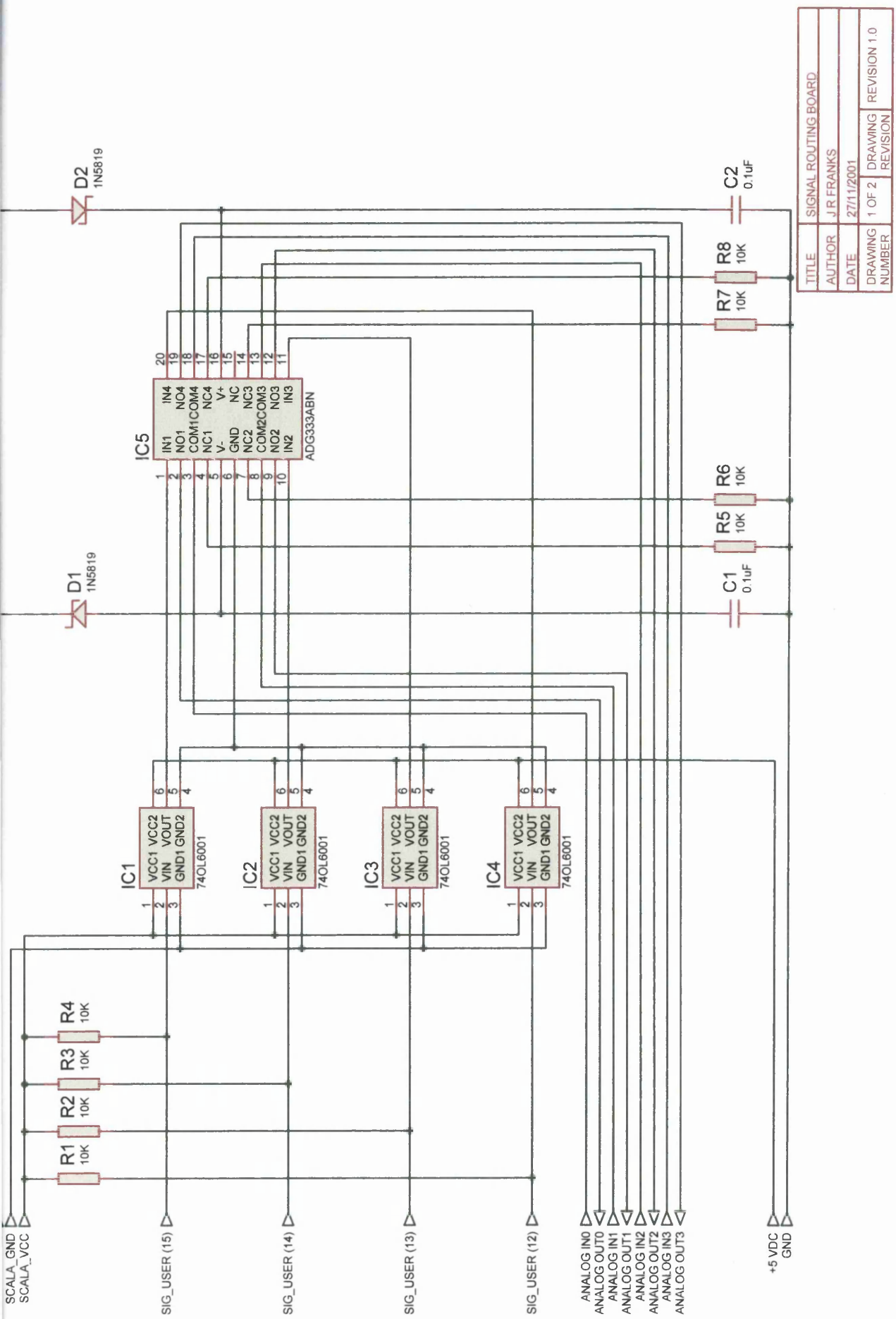


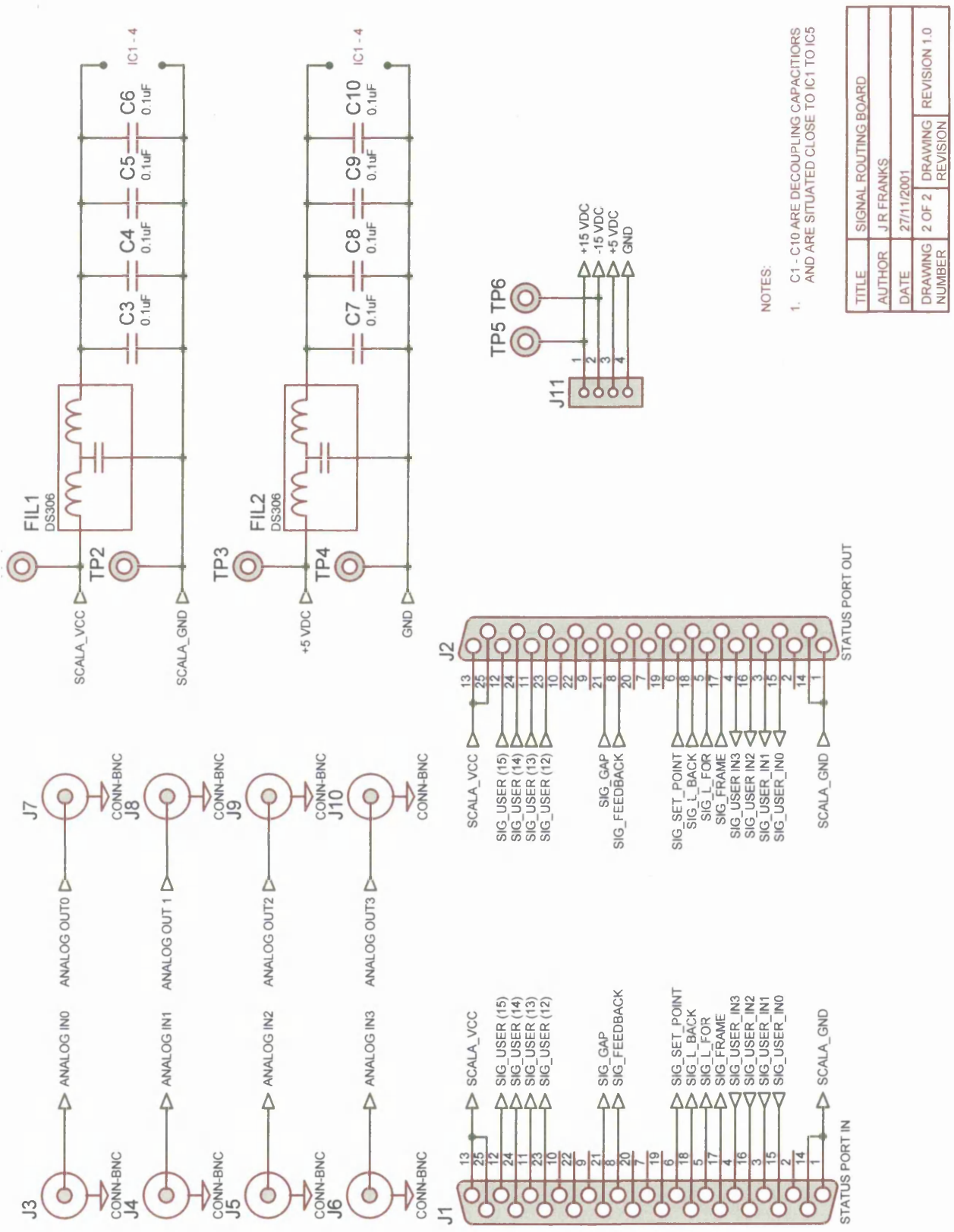
Figure 4.6.2: Functional block diagram of the Signal Routing module circuitry.

The Signal Routing module is responsible for independently switching up to four signals under control of the SCALA electronics status port. An Analog Devices quad integrated switch, comprising of four SPDT type CMOS switches that are independently controllable, forms the basis of the switching circuitry. Opto-isolators in the digital signal paths between the status port and the analog switch minimise contamination of the switched signals with noise. Fig 4.6.1 shows a photo of the assembled Signal Routing module complete with power supply encased in 1U rack case. Fig 4.6.2 shows the functional block diagram for the Signal Routing module. A detailed description of each functional unit in Fig 4.6.2 is presented in section 4.6.2 with reference to the circuit schematic for the signal switching module, Fig 4.6.3.

4.6.1 Description of the Signal Routing module circuitry.

In the circuit schematic shown in figure 4.6.3, opto-isolators IC1 to IC4 isolate status port signals 15 to 12 from control inputs IN1 to IN4 of IC5 respectively. Status port signals 15 to 12 are pulled up to the SCALA electronics Vcc power rail with 10K resistors R1 to R4, ensuring defined logic levels for IC1 to IC4 transmitters in the absence of an output from the status port. IC1 to IC4 are similar to devices used in the DAC module with the exception that logically they are inverters as opposed to buffers [7].


D2
1N5819
D1
1N5819
C2
0.1uF
C1
0.1uF
R8
10K
R7
10K
R6
10K
R5
10K
R4
10K
R3
10K
R2
10K
R1
10K
IC5
ADG333ABN
IC1
IC2
IC3
IC4
74OL6001
TITLE
SIGNAL ROUTING BOARD
AUTHOR
J R FRANKS
DATE
27/11/2001
DRAWING NUMBER
1 OF 2
DRAWING REVISION
REVISION 1.0
SCALA_GND
SCALA_VCC
SIG_USER (15)
SIG_USER (14)
SIG_USER (13)
SIG_USER (12)
ANALOG IN0
ANALOG OUT0
ANALOG IN1
ANALOG OUT1
ANALOG IN2
ANALOG OUT2
ANALOG IN3
ANALOG OUT3
+5 VDC
GND


FIL1
DS306
FIL2
DS306
TP2
TP3
TP4
TP5
TP6
SCALA_VCC
SCALA_GND
+5 VDC
GND
C3 0.1uF
C4 0.1uF
C5 0.1uF
C6 0.1uF
C7 0.1uF
C8 0.1uF
C9 0.1uF
C10 0.1uF
IC1 - 4
J11
+15 VDC
-15 VDC
+5 VDC
GND
J1
J2
J3
J4
J5
J6
J7
J8
J9
J10
CONN-BNC
ANALOG IN0
ANALOG IN1
ANALOG IN2
ANALOG IN3
ANALOG OUT0
ANALOG OUT 1
ANALOG OUT2
ANALOG OUT3
SCALA_VCC
SIG_USER (15)
SIG_USER (14)
SIG_USER (13)
SIG_USER (12)
SIG_GAP
SIG_FEEDBACK
SIG_SET_POINT
SIG_L_BACK
SIG_L_FOR
SIG_FRAME
SIG_USER_IN3
SIG_USER_IN2
SIG_USER_IN1
SIG_USER_IN0
SCALA_GND
STATUS PORT IN
STATUS PORT OUT
NOTES:
1. C1 - C10 ARE DECOUPLING CAPACITORS AND ARE SITUATED CLOSE TO IC1 TO IC5
TITLE SIGNAL ROUTING BOARD
AUTHOR J R FRANKS
DATE 27/11/2001
DRAWING NUMBER 2 OF 2
DRAWING REVISION REVISION 1.0

IC5 is an ADG333A [8] containing four independently controllable SPDT CMOS switches. The control inputs IN1 to IN4 operate switches 1 to 4 respectively, routing a signal from its respective common (COM1 to COM4) to either a normally closed (NC1 to NC4) or a normally open output (NO1 to NO4). Application of a TTL compatible logic 0 voltage to a control input results in routing of the input signal to the normally closed output. Conversely, a TTL compatible logic 1 voltage routes the signal to the normally open output. Since IC1 to IC4 perform logical inversion in addition to optical isolation, logically the operation viewed from the transmitters of IC1 to IC4 is the compliment of the above argument. Resistors R5 to R8 provide signal termination of the normally closed outputs of IC5; their values selected to match the typical input impedance of the SCALA electronics external inputs. Schottky diodes D1 and D2 prevent potentially fatal 'latch-up' of the CMOS switches during power up. Murata 3 terminal filters FIL1and FIL2 and decoupling capacitors C3 to C8 placed in both 5VDC supplies provide attenuation of high frequency noise. Decoupling capacitors C1 and C2 perform a similar function in the +/- 15VDC supplies to IC5.

Care has been taken in the layout of the PCB to ensure isolation of signal paths to minimise cross coupling of signals. PCB artwork for the Signal Switching module is located in appendix A.

4.7 Power Supply

Two separate target boards with differing requirements are accommodated by the power supply. The supply is split into two distinct sections termed 'analog' and 'digital'. The analog supply is required to provide both positive and negative output voltages at low currents (≤ 100mA). To accommodate both target boards the outputs are manually adjustable. The digital supply provides a single output voltage that meets TTL device supply requirements. The supplies are kept electrically, and to an extent physically separate, to avoid coupling of noise inherent in digital circuitry introducing undesirable electrical noise into the analog supply. Fig 4.7.1 shows a photo of an assembled power supply board and ancillary components installed in the DAC module.

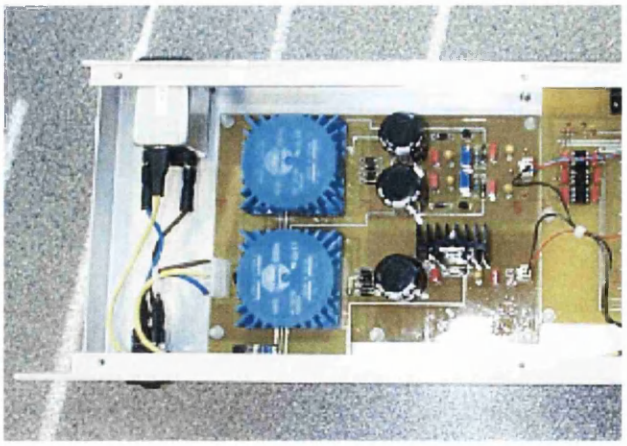
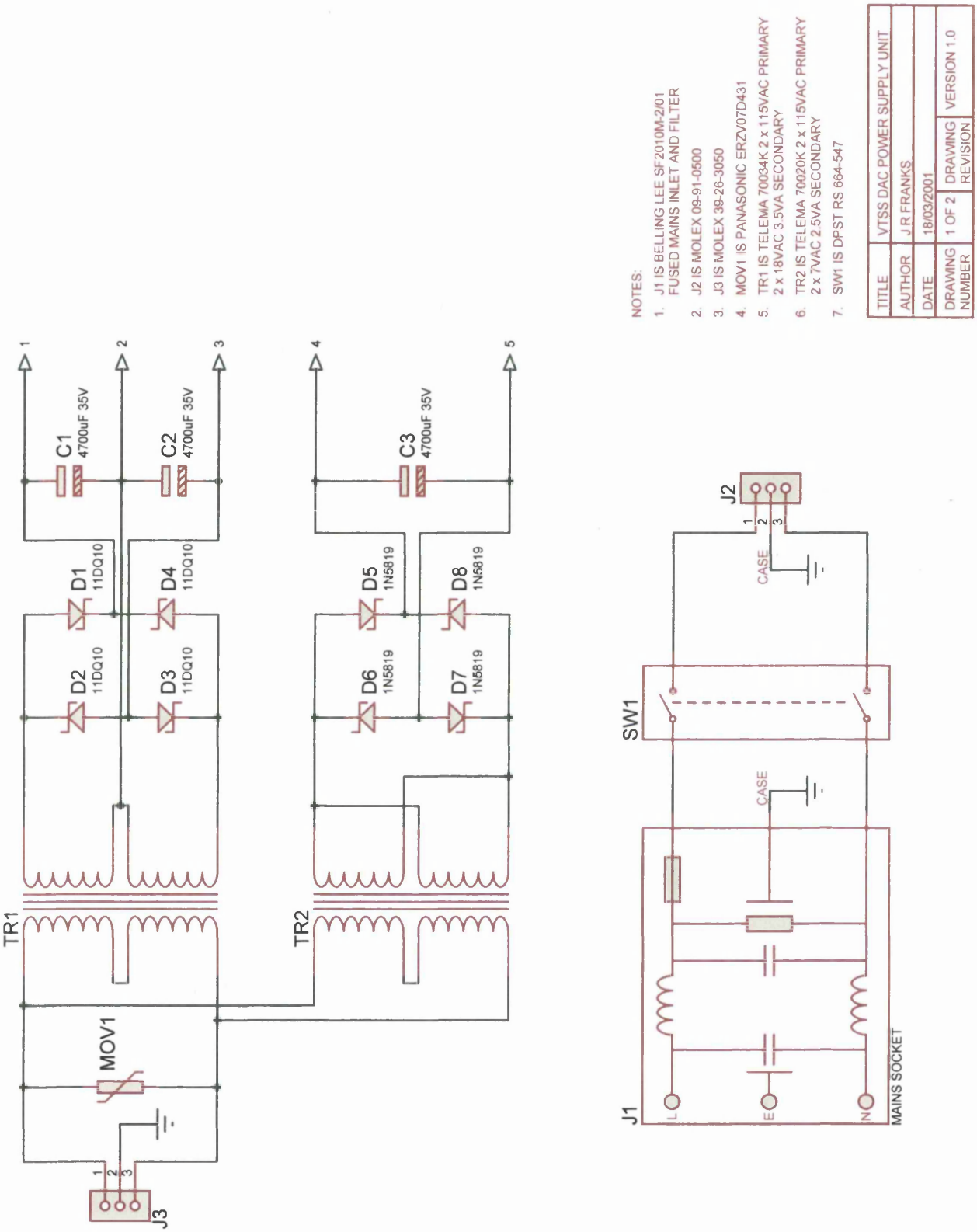


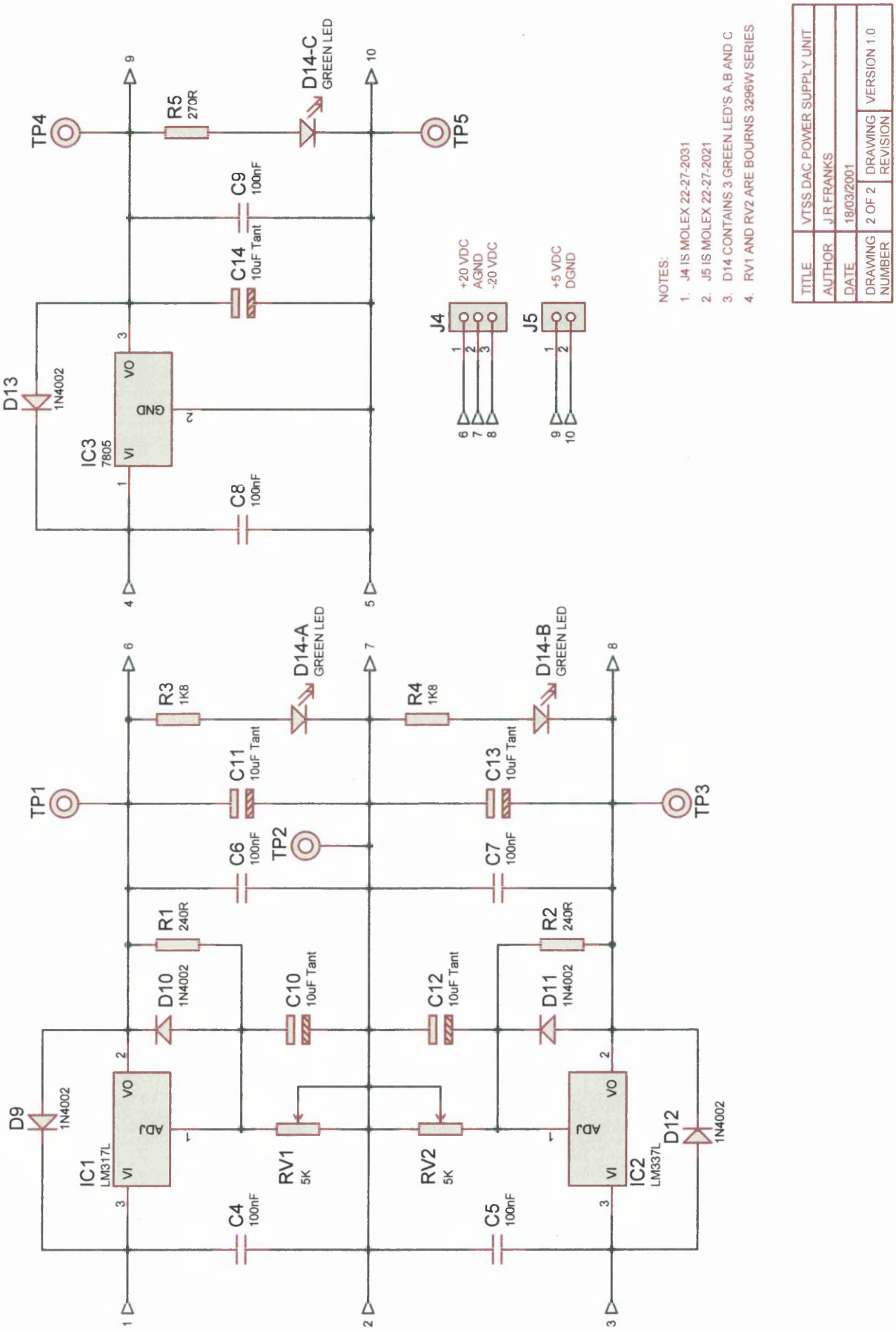
Figure 4.7.1: The power supply unit installed in the DAC module.

The majority of power supply components are accommodated on a custom designed PCB (see appendix A), with the exception of a mains inlet / filter and switch. Mains inlet fuse / filter combination J1 (Fig 4.7.1 upper left corner) is included as a preventative measure, to reduce the likelihood of transmission of mains born noise to the supply outputs. Both neutral and live supplies are switched by SW1 (Fig 4.7.1 lower left corner), ensuring the power supply PCB can be fully isolated from the mains. Connections to the PCB are made using two part connectors with appropriate specifications. Effort has been made to ensure the grounds of analog and digital sections remain separate on the PCB. Star grounding techniques are then employed where the grounds of the supplies meet, ensuring analog and digital supply grounds are at the same potential. Discussion of specifics relating to analog and digital supplies follows with reference to the circuit schematic Fig 4.7.2.

4.7.1 The analog supply.

The two 115V_{AC} primary windings of TR1 are connected in series to effectively create a single 230V_{AC} winding. Transient suppression is provided by connecting a metal-oxide varistor in parallel with the primaries of TR1 and TR2 [9]. Connecting the two separate secondary windings of TR1 in series effectively creates a centre tapped secondary winding; the tap formed at the node where the two windings meet. Schottky rectifier diodes D1 to D4 can be considered as two separate full wave rectifier circuits, whose equivalent circuits are shown in Fig 4.7.3.


NOTES:
1. J1 IS BELLING LEE SF2010M-2/01 FUSED MAINS INLET AND FILTER
2. J2 IS MOLEX 09-91-0500
3. J3 IS MOLEX 39-26-3050
4. MOV1 IS PANASONIC ERZV07D431
5. TR1 IS TELEMA 70034K 2 x 115VAC PRIMARY 2 x 18VAC 3.5VA SECONDARY
6. TR2 IS TELEMA 70020K 2 x 115VAC PRIMARY 2 x 7VAC 2.5VA SECONDARY
7. SW1 IS DPST RS 664-547
TITLE VTSS DAC POWER SUPPLY UNIT
AUTHOR J.R.FRANKS
DATE 18/03/2001
DRAWING NUMBER 1 OF 2
DRAWING REVISION VERSION 1.0
C1 4700uF 35V
C2 4700uF 35V
C3 4700uF 35V
D1 11DQ10
D2 11DQ10
D3 11DQ10
D4 11DQ10
D5 1N5819
D6 1N5819
D7 1N5819
D8 1N5819
TR1
TR2
MOV1
J3
J2
CASE
SW1
J1
L
E
N
MAINS SOCKET


TP4
R5
270R
D14-C
GREEN LED
TP5
C9
100nF
C14
10uF Tant
D13
1N4002
IC3
7806
VO
GND
VI
C8
100nF
J4
+20 VDC
AGND
-20 VDC
J5
+5 VDC
DGND
NOTES:
1. J4 IS MOLEX 22-27-2031
2. J5 IS MOLEX 22-27-2021
3. D14 CONTAINS 3 GREEN LED'S A,B AND C
4. RV1 AND RV2 ARE BOURNS 3296W SERIES
TITLE
VTSS DAC POWER SUPPLY UNIT
AUTHOR
J.R FRANKS
DATE
18/03/2001
DRAWING NUMBER
2 OF 2
DRAWING REVISION
VERSION 1.0
D14-A
GREEN LED
D14-B
GREEN LED
R3
1K8
R4
1K8
C11
10uF Tant
C13
10uF Tant
TP1
TP2
TP3
C6
100nF
C7
100nF
R1
240R
R2
240R
D10
1N4002
C10
10uF Tant
C12
10uF Tant
D11
1N4002
D9
1N4002
IC1
LM317L
ADJ
RV1
5K
RV2
5K
IC2
LM337L
D12
1N4002
C4
100nF
C5
100nF

D1 and D4 rectify the positive half of the supply and D2 and D3 the negative. Schottky rectifier diodes are used in preference to their silicon counterparts to minimize RF production, which is a side effect of the charge storage mechanism inherent in silicon rectifiers [10].

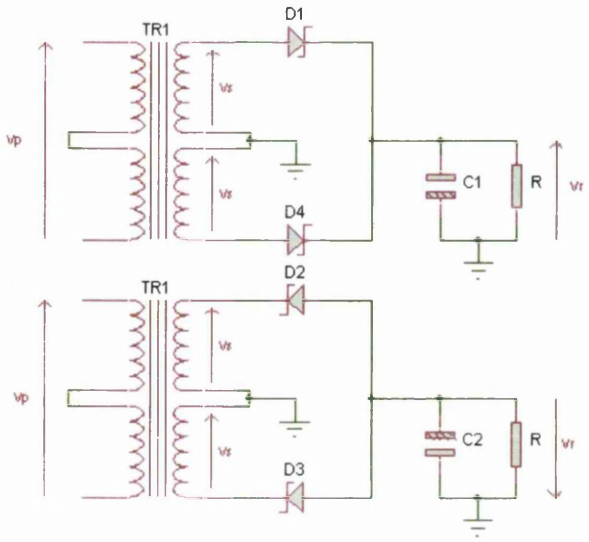


Figure 4.7.3: Equivalent circuits for both full wave rectifiers. Vp is the primary voltage, Vs the secondary voltage and Vr the voltage across the equivalent load resistance R.

Filtration of the raw DC is achieved by C1 for the positive supply and C2 for the negative. The values of C1 and C2 were selected to ensure the ripple voltage prior to regulation did not exceed 100m$V_{PK\text{-}PK}$.

$$Vr = \sqrt{2}Vs - Vf \tag{4.1}$$

The voltage (Vr) appearing across the equivalent load resistance R depends on the load current drawn from the secondary winding of TR1. To specify minimum and maximum load voltages TR1 is considered in the fully loaded and un-loaded conditions. The secondary voltage (Vs) is specified as 18V_{RMS} fully loaded, rising to 1.35Vs un-loaded. Applying equation 4.1 with the maximum forward voltage (Vf) of D1 to D4 as 0.85V_{DC} yields the load voltage as 24.6V_{DC} fully loaded, rising to 33.4V_{DC} un-loaded. Since TR1 is never subject to fully loaded conditions under normal operation, the maximum required output voltage after regulation (+/- 20V_{DC}) can be maintained. Theodore and Bogart [11] present an analysis of rectifier circuit

topologies and filters, to which the reader is referred for a more comprehensive understanding.

$$Vout = Vref\left(1+\frac{RVx}{Rx}\right)+IadjRVx \qquad (4.2)$$

Linear adjustable voltage regulators IC1 and IC2 [12], [13] (LM317LZ and LM337LZ respectively) regulate filtered positive and negative supplies to defined output voltages. In both cases the output voltage is defined by a pair of resistors (R1 and RV1 for IC1 and R2 and RV2 for IC2) in accordance with equation 4.2. The manufacturer defines the parameters *Vref* and *Iadj* in equation 4.2. RV1 and RV2 are incorporated to facilitate a range of output voltages to be accommodated in accordance with the requirements of the target board.

Capacitors C10 and C12 in figure 4.7.2 provide low impedance paths to noise appearing on the adjustment terminals of the regulators, ensuring the adjustment reference voltage remains stable. Addition of protection diodes D9 – D12 ensures protection against input and output short circuits. Capacitors C4 and C5 provide input bypassing to improve ripple rejection characteristics of IC1 and IC2. C6, C11, C7 and C13 provide additional filtering and improvements in both output impedance and transient response. The combination of R3, R4, D14-A and D14-B provide visual indication that the analog supply is operating. Resistors R3 and R4 provide current limiting for D14-A and B respectively.

4.7.2 The digital supply.

The two 115V_{AC} primary windings of TR2 are connected in series to effectively create a single 230V_{AC} winding. The two secondary windings of TR2 are connected in parallel to effectively create a single winding with double the current rating. Schottky diodes D5-D8 are connected in a standard bridge topology [11], providing full wave rectification. To specify minimum and maximum load voltages as with TR1, TR2 is considered in the fully loaded and un-loaded conditions. The secondary voltage is specified as 7V_{RMS} fully loaded, rising to 1.4Vs un-loaded. Applying equation 4.3 with the maximum forward voltage (Vf) of D5 to D8 as 0.85V_{DC} yields the load

voltage as 8.2VDC fully loaded, rising to 12.2VDC un-loaded. C3 provides filtration of the raw DC.

$$Vr = \sqrt{2}Vs - (2Vf) \tag{4.3}$$

IC3 (LM7805) [14] is a fixed output voltage linear regulator that regulates the filtered supply to 5VDC. Adequate heat sinking of IC3 is provided to ensure reliable operation at higher currents. D3 is incorporated to provide input short circuit protection. C8 provides input bypassing to provide good ripple rejection characteristics. C14 and C9 provide additional filtering and improvements in both output impedance and transient response. The combination of R5 and D14-C provide visual indication that the supply is operating.

4.8 The attenuator module.

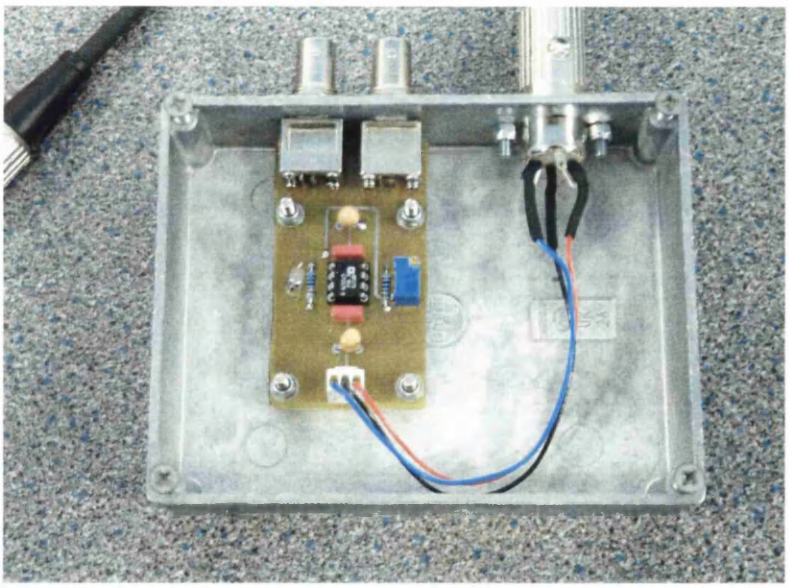
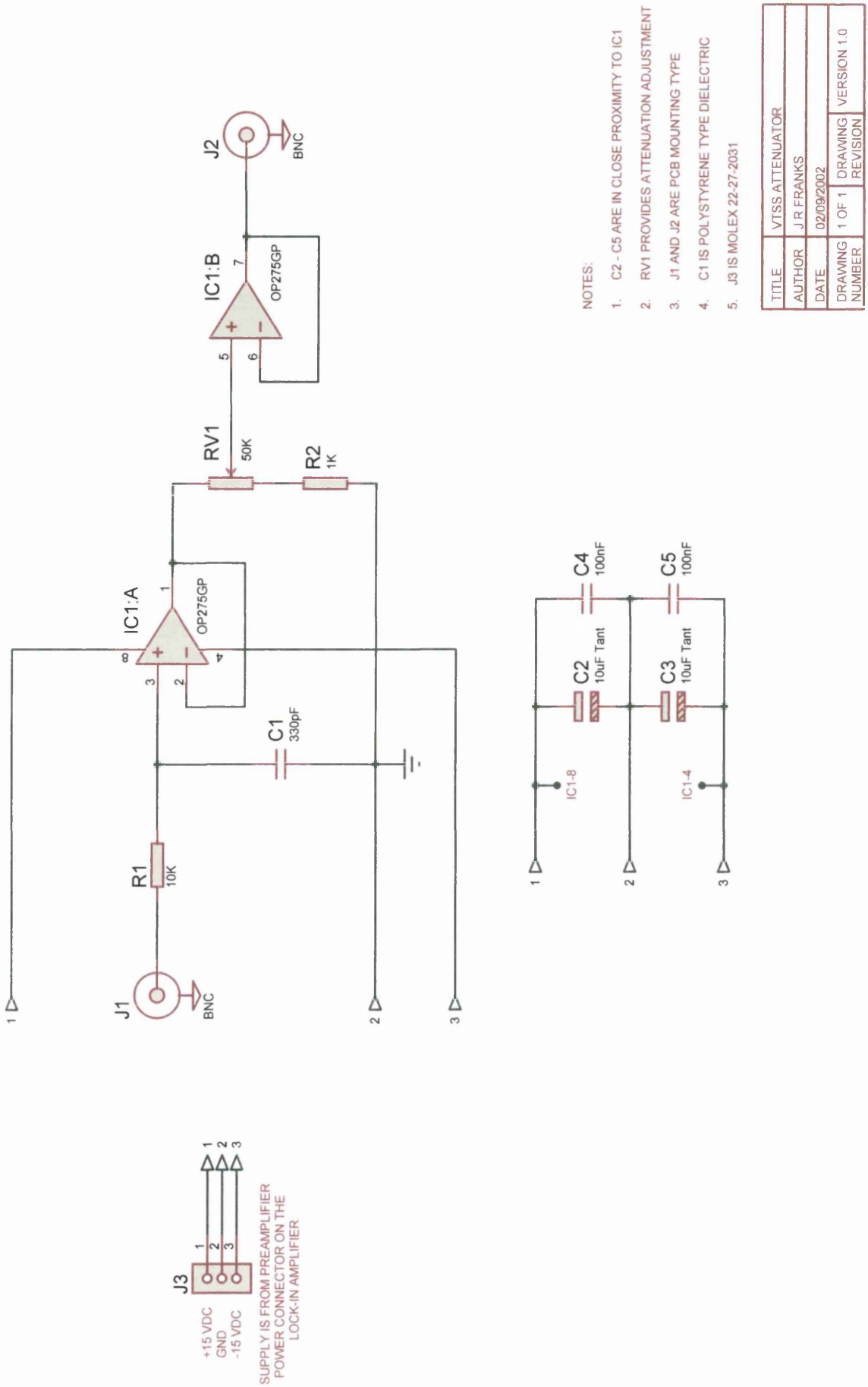


Figure 4.8.1: A picture of the attenuator module encased within an aluminium die cast enclosure.

The attenuator is responsible for scaling the output voltage from the SCALA electronics tunnel current monitor output before applying it to the Lock-in amplifier. The SCALA manual specifies an output of 1V for a 5nA tunnel current, however in practice measurements show 1.28V represents a tunnel current of 5nA. Therefore, the


J2
BNC
IC1:B
OP275GP
7
5
6
RV1
50K
R2
1K
IC1:A
OP275GP
1
8
4
3
2
C1
330pF
R1
10K
J1
BNC
1
2
3
C4
100nF
C5
100nF
C2
10uF Tant
C3
10uF Tant
IC1-8
IC1-4
NOTES:
1. C2 - C5 ARE IN CLOSE PROXIMITY TO IC1
2. RV1 PROVIDES ATTENUATION ADJUSTMENT
3. J1 AND J2 ARE PCB MOUNTING TYPE
4. C1 IS POLYSTYRENE TYPE DIELECTRIC
5. J3 IS MOLEX 22-27-2031
TITLE
VTSS ATTENUATOR
AUTHOR
J R FRANKS
DATE
02/09/2002
DRAWING NUMBER
1 OF 1
DRAWING REVISION
VERSION 1.0
J3
+15 VDC
GND
-15 VDC
SUPPLY IS FROM PREAMPLIFIER
POWER CONNECTOR ON THE
LOCK-IN AMPLIFIER

attenuator provides a convenient way of correcting for the inaccuracy of the tunnel current monitor output. Additionally, to facilitate higher Lock-in input sensitivities the attenuation can be increased, thus enabling measurements on less conductive samples to be performed. Figure 4.8.1 shows a picture of the attenuator module encased in aluminium die cast enclosure.

The attenuator circuit Fig 4.8.2 consists of a dual operational amplifier, input filter network and attenuator network. R1 and C1 form a low pass filter network with a 3db cut off frequency of 48KHz. The 6db per octave roll-off of the low pass filter ensures that high frequency noise is attenuated. IC1-A is configured as a voltage follower whose output feeds the attenuator network RV1 and R2. Equation 4.4 defines the output voltage, Vout from the attenuator network in terms of the input voltage, Vin.

$$Vout = Vin\left(\frac{R2}{RV1 + R2}\right) \tag{4.4}$$

Considering RV1 set to its maximum value yields Vout = Vin, whereas when set to its minimum value Vout = 1/51Vin. IC1-B is configured as voltage follower, whose function is to buffer the output from the attenuator network. Decoupling of the supplies to IC1 is performed by C2 – C5. PCB layouts for the attenuator can be found in appendix A.

4.9 The Perkin Elmer DSP Lock-in amplifier.

The Lock-in amplifier is responsible for measurement of the sample conductivity at each discrete step in sample bias during variable tip-sample separation STS. A Perkin Elmer model 7265 DSP lockin amplifier [15], Fig 4.9.1 is used to accomplish this task.

The Lock-in amplifier effectively differentiates the tunnel current with respect to the modulating voltage, producing a steady state output voltage whose value is proportional to the conductivity. The term 'steady state' output voltage is used because the output voltage requires a period of time to settle to its final value or steady state. The time period is defined by selection of a Lock-in time constant.

Figure 4.9.1: The Perkin Elmer model 7265 DSP Lock-in amplifier.

A sinusoidal reference from the internal oscillator of the Lock-in is used to modulate the sample bias for a specific period of time, during each step in sample bias. The time period depends on the selection of Lock-in time constant. The modulation of the sample bias produces a corresponding sinusoidal change in the tunnel current at the same frequency. The SCALA electronics tunnel current monitor output produces an output voltage proportional to the tunnel current, 1nA being represented by 200mV. This output is attenuated for reasons as discussed in 4.8 and fed to the single ended input of the Lock-in. Selection of the input sensitivity depends on the conductivity of the sample under investigation. Samples with higher conductivities require a reduction in the input sensitivity, whereas samples with lower conductivity require higher sensitivities. The overall sensitivity is not solely determined by the Lock-in amplifier, but by a combination of the Lock-in sensitivity and the attenuation of the attenuator. The output voltage from the Lock-in is logged through use of one of the SCALA electronics external ADC inputs, therefore enabling the conductivity to be measured and recorded for each discrete step in sample bias.

4.10 Printed circuit board design and manufacture.

Here a description of the design and manufacture process used to produce the printed circuit boards as shown in appendix A is detailed. A description of the methodology

employed during board design is discussed in section 4.10.1, with reference to the 'Ares Lite' software package. The manufacture process employed in the production of the boards is then detailed in section 4.10.2.

4.10.1 The Ares software package and board design methodology.

The software package used to develop PCB artworks is 'Ares Lite' [16], a specialist package manufactured by 'Labcenter Electronics LTD'. Ares Lite is a simplified version of an industrial package and omits such features that are desirable and necessary in industry, however sufficient features are retained to allow production of high quality artwork. Figure 4.10.1 shows a screen shot of the Ares environment with a section of the power supply board displayed in the design window.

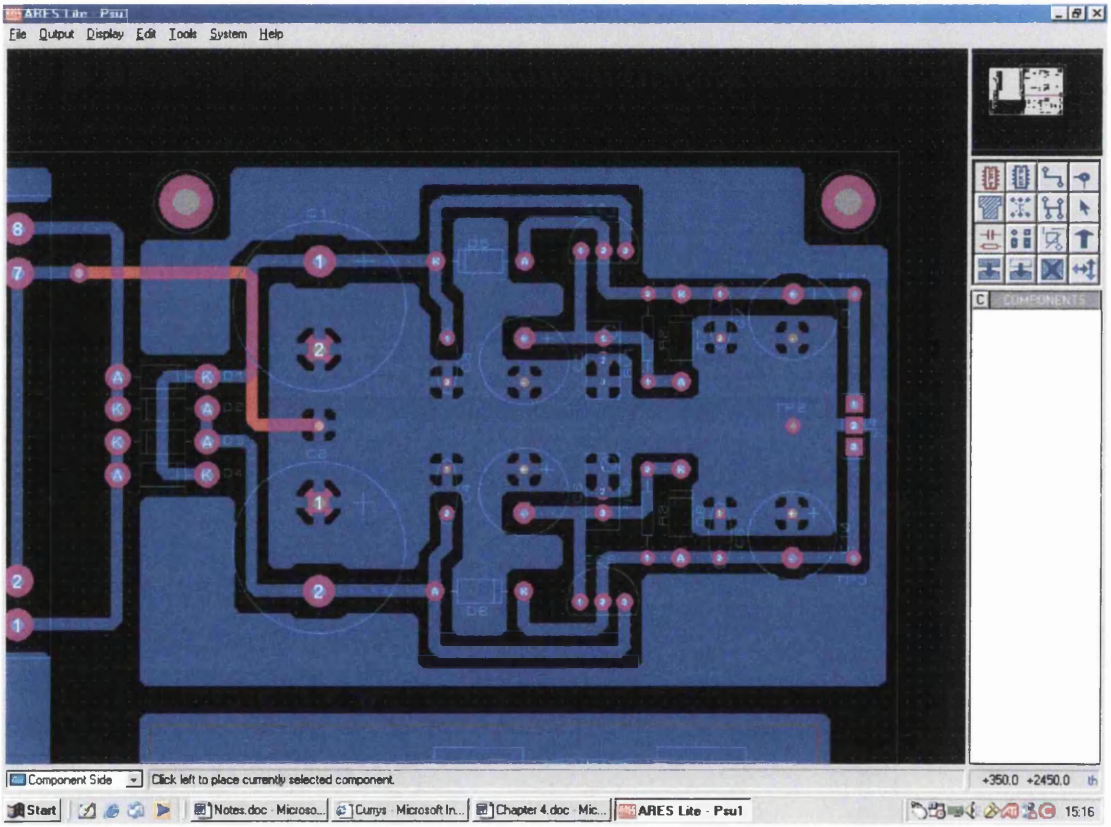


Figure 4.10.1:A screen shot of the Ares environment with a section of the power supply board displayed in the design window.

The first task undertaken in a new design is to mechanically layout the board by defining the board size and fixing positions. Components are then added and positioned dependent on mechanical and electrical constraints specific to the circuit. Individual components not contained within Ares libraries can be designed from primitives and added to the design. Specific circuit nodes are connected with tracking whose size and positioning is governed by the currents and signals the tracking is required to carry. Vias connecting tracking layers are added and low impedance ground and power planes.

4.10.2 The board manufacture process.

Completed artwork for both top and bottom tracklayers is printed onto acetate film with an inkjet printer, creating a mask-set. The acetate films are aligned and fixed together, and a piece of photosensitised copper clad board sandwiched in between. UV light is used to expose the board, which chemically alters unmasked areas in the photosensitive coating. A bath of heated sodium hydroxide is used to wash away the exposed photosensitive coating leaving the masked areas intact. The board is transferred into a heated bath of ferric chloride that etches away the unmasked copper areas. Following the etching process the board is washed to remove the remaining photosensitive coating and plated with tin to prevent the copper tracks oxidising. The board is drilled to suit component requirements, and via pins soldered in position to connect copper layers.

Components are fitted to the board in size order and soldered. The complete assembly is washed with isopropanol to remove remnants of flux from the soldering process and finally coated with a clear protective lacquer.

4.11 References.

[1] R.M. Feenstra, J.A. Stroscio, 'Methods of Experimental Physics: Scanning Tunnelling Microscopy', Chapter 4, Edit J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[2] Omicron NanoTechnology, 'The SCALA PRO Software Manual', Version 4.1 (2001).

[3] Motorola, M68HC12B Family Technical Data, Rev 4.0.

[4] Motorola,CPU12 Reference Manual, Rev 3.0.

[5] P&E Microsystems web site, PEMICRO.COM.

[6] Analog Devices, AD669 data sheet, Rev.A.

[7] Quality Technologies, 74OL6000,74OL6001,74OL6010 and 74OL6011 data sheet.

[8] Analog Devices, ADG333A data sheet, Rev. 0.

[9] I. M. Gottlieb, Regulated Power Supplies, fourth edition, **68 – 73**, TAB Books, 1992.

[10] I. M. Gottlieb, Regulated Power Supplies, fourth edition, **134 – 141**, TAB Books, 1992.

[11] F. Theodore and J. R. Bogart, Electronic Devices And Circuits, second edition, Chapter 16, Merril publishing company, 1991.

[12] National Semiconductor, LM317 data sheet.

[13] National Semiconductor, LM337 data sheet.

[14] National Semiconductor, LM78XX series data sheet.

[15] Perkin Elmer, Model 7265 DSP Lock-in Amplifier, instruction manual, 2000.

[16] Labcenter Electronics web site, LABCENTER.CO.UK

Chapter 5

Software Development

Introduction

Herein follows a discussion of the development and function of software specifically created for the variable tip-sample separation STS application. The software is comprised of three distinct sections: code developed for the MC68HC12 microcontroller, the graphical user interface developed for the PC platform and macros developed within the SCALA Pro nano-structuring language.

The microcontroller code development is discussed in sections 5.1 and 5.2. In 5.1 the algorithms employed for both linear and non-linear programs are examined from the perspective of implementation on the microcontroller platform. In 5.2 the P&E Microsystems integrated design environment, employed in code development is introduced and the microcontroller application code dissected and examined in detail. Discussion of the graphical user interface is undertaken in section 5.3, proceeding with an introduction to Borland Delphi and then an examination of the functional operation of the interface. Section 5.4 introduces macros written within the SCALA Pro nano-structuring language to perform both hardware and software based variable tip-sample separation STS.

5.1 Linear and Non-Linear program algorithms.

Here a description of the algorithms used to entirely implement linear and partially implement non-linear variable tip-sample separation STS solutions is given. The linear algorithm is discussed in section 5.1.1 and the non-linear algorithm in section 5.1.2.

5.1.1 Linear algorithm.

The linear algorithm implemented by the microcontroller manipulates tip-sample separation linearly during spectroscopic measurements, such that as the sample bias approaches zero from either maximum, the tip-sample separation is linearly reduced until the sample bias reaches zero, at which point the tip-sample separation is linearly increased.

During acquisition of an I/V spectra containing n data points, where n is an odd integer between 1 and 255, for the first (n-1)/2 data points, the tip sample separation is reduced at each data point by a quantity defined as *Step Size* by equation 5.1. At data point ((n-1) /2) + 1 the tip has reached its maximum displacement from the regulated height. For the remaining (n-1)/2 data points the tip sample separation is increased by *Step Size* for each data point. The *Step Size* as defined by equation 5.1 is calculated by dividing $0CCC, which is the hexadecimal equivalent of 0.1nm tip movement by (n-1)/2 and multiplying the result by an integer *Scale Factor* in the range 1 – 9. Multiplying by the *Scale Factor* sets the peak tip displacement achieved from the regulated height. Figure 5.1.1 illustrates diagrammatically the operation of the linear algorithm.

$$Step\ Size = \left(\frac{\$0CCC}{(n-1)/2} \right) \times Scale\ Factor \tag{5.1}$$

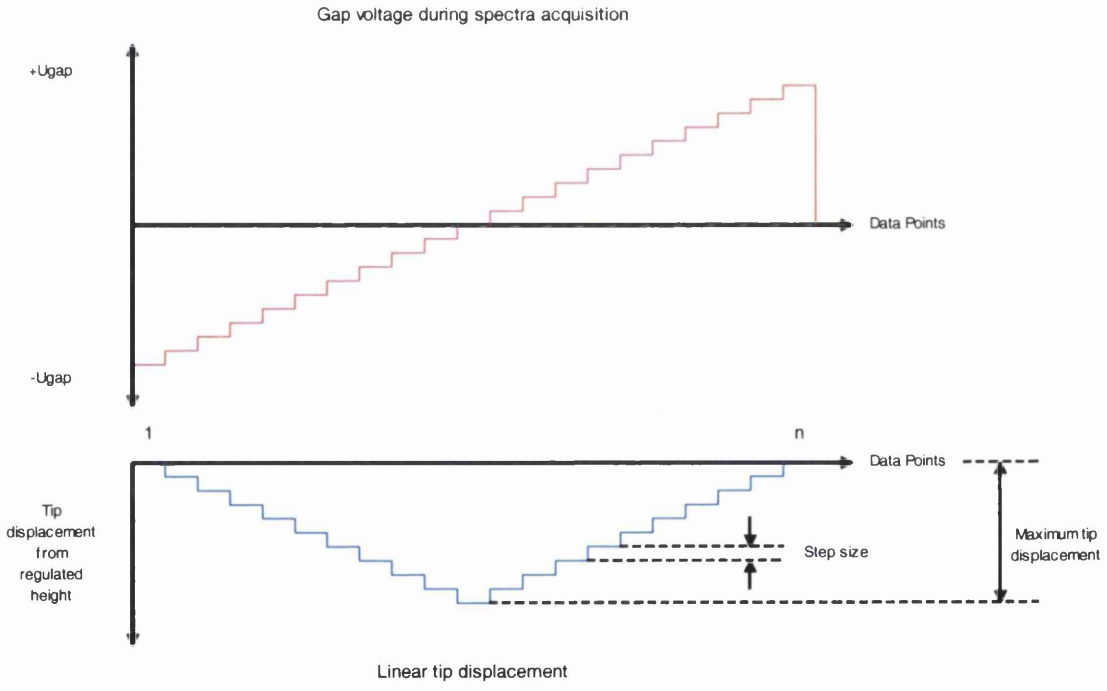


Figure 5.1.1: Diagrammatic illustration of the operation of the linear algorithm.

5.1.2 Non-Linear algorithm.

The non-linear algorithm as implemented by the microcontroller is a partial realization of a complete solution. Functionally the portion of the solution realized measures a signal presented to the microcontrollers analog to digital converter and interrogates a look-up table, returning a value used in the calculation of the output.

$$Step\ Size = \frac{Max\ Step\ Size}{\$FF} \tag{5.2}$$

To calculate the look-up table a quantity defined as *Max Step Size*, which is the maximum allowed tip movement for a single data point is divided by $FF, generating a quantity defined as *Step Size* by equation 5.2. The look-up table contains 256 entries; the first entry is $0000 and the last entry Max Step Size. Each entry within the look-up table is calculated by adding *Step Size* to the previous entry, starting with the second entry.

For a measured spectrum containing n data points, and for the first (n-1)/2 data points, the value returned from interrogation of the look-up table is added to an accumulator. If the value contained within the accumulator exceeds a limit defined as Z Limit the tip displacement from the regulated height is limited to Z Limit. However, if the accumulator value is < Z Limit the tip displacement follows the value of the accumulator. For the next (n-1)/2 data points the value returned from interrogation of the look-up table is subtracted from the accumulator. If the value contained within the accumulator exceeds Z Limit the tip displacement form the regulated height is limited to Z Limit. However, if the accumulator value is < 0 the tip displacement is 0. When the accumulator value is < Z limit and > 0 the tip displacement follows the accumulator. For the last data point the tip displacement is 0 irrespective of the contents of the accumulator.

5.2 Microcontroller code and code development environment.

5.2.1 P&E Microsystems integrated design environment for the MC68HC12 family.

The P&E Microsystems integrated design environment (IDE) for the MC68HC12 family of microcontrollers was used during the development and verification of application code. The IDE is modular in its construction and consists of a text editor, assembler, programmer and debugger. Figure 5.2.1 shows a screen shot of the IDE with an editor window open.

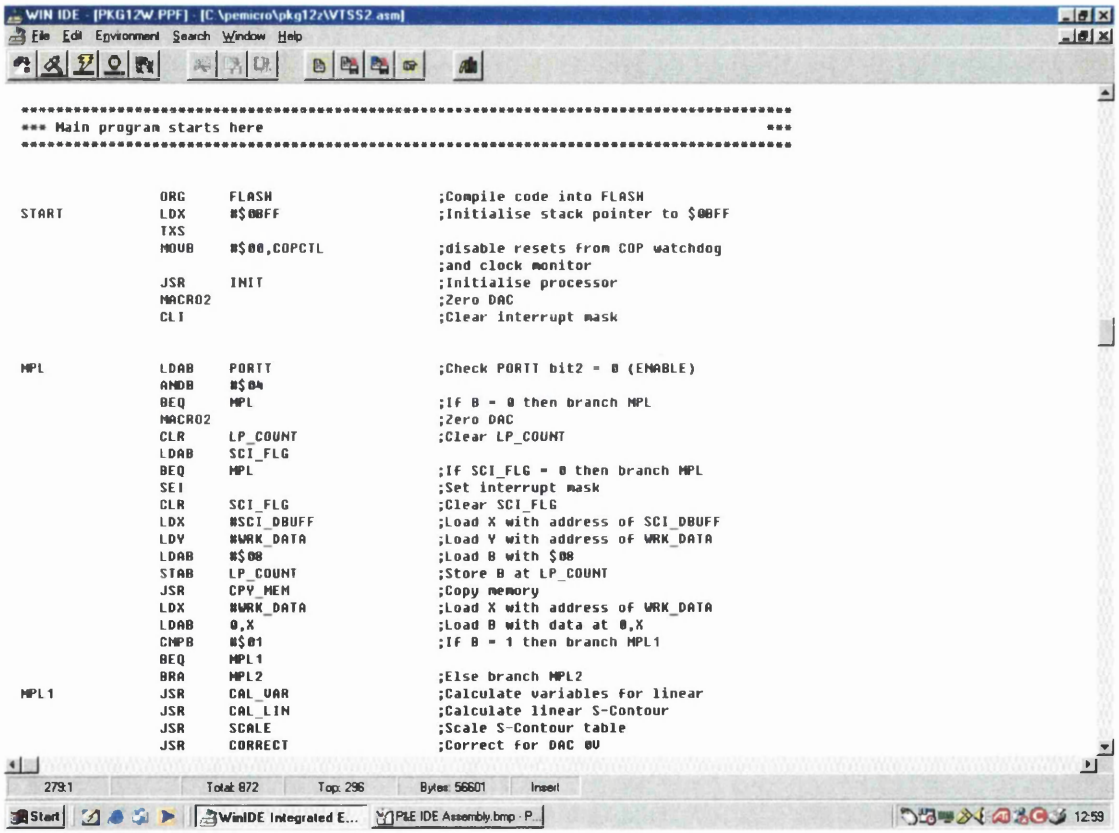


Figure 5.2.1: A screen shot illustrating the text editor window.

When the application code is complete, it is assembled into a format compatible with the microcontroller. During the assembly process the text of the application code is checked for syntactical errors and translated into MC68HC12 machine code. Figure 5.2.2 shows a screen shot illustrating the assembler window.

Following successful assembly, the machine code file generated by the assembler is loaded into the microcontroller's internal memory by use of the programmer. The microcontroller implements a background debug interface [1] through which the code is downloaded. Figure 5.2.3 shows a screen shot illustrating the programmer window. Following successful programming the code within the microcontroller can be evaluated in terms of its function by use of the debugger via the background debug interface. The background debug interface allows execution of single instructions within the application code, whist monitoring the microcontroller's memory and register status, therefore providing a convenient and powerful medium through which the application code can be evaluated.

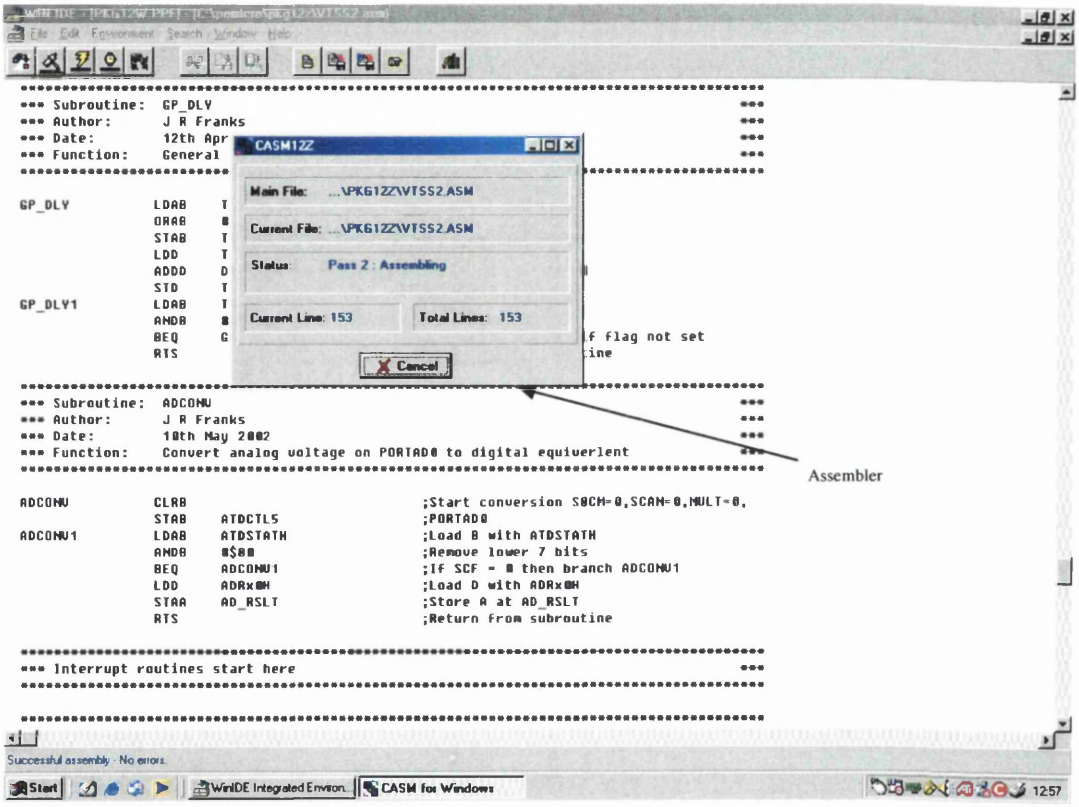


Figure 5.2.2: Screen shot illustrating the assembler window.

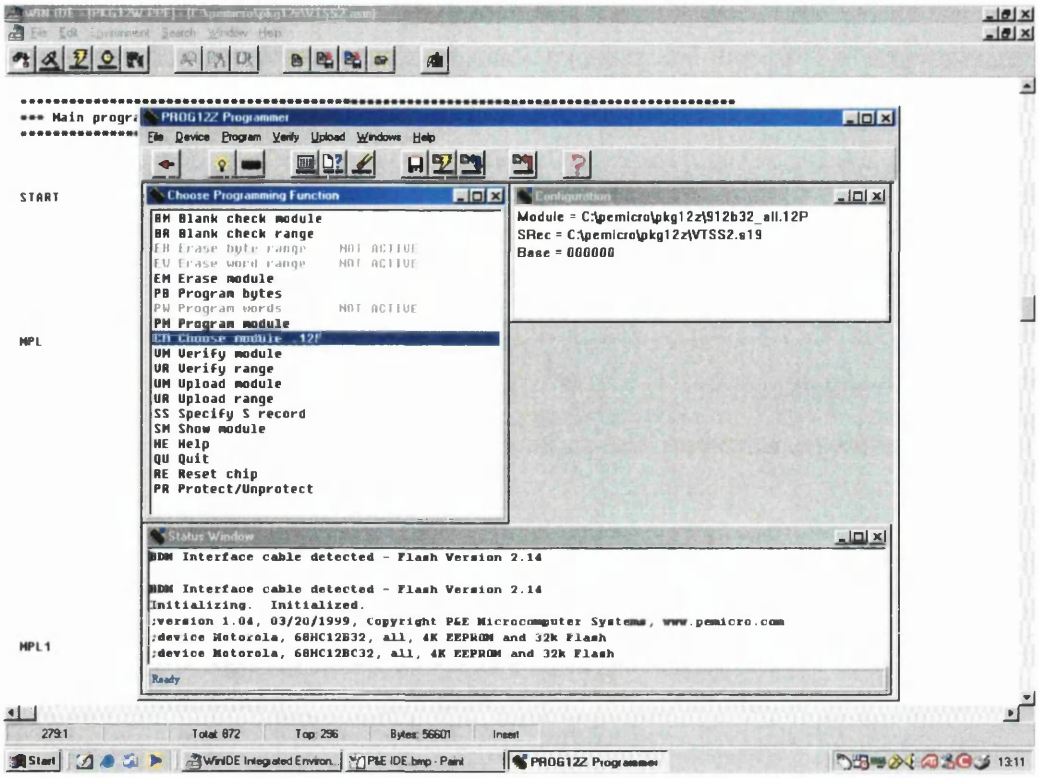


Figure 5.2.3: Screen shot illustrating the programmer window.

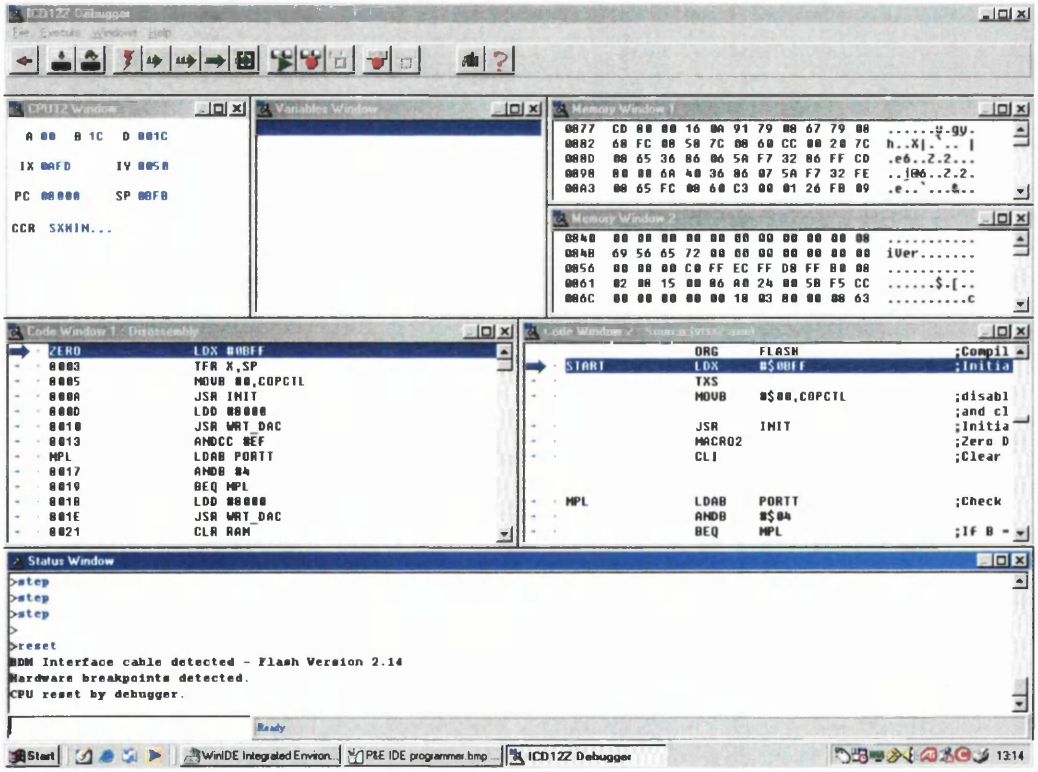


Figure 5.2.4: Screen shot illustrating the debugger window with sub-windows labelled.

Figure 5.2.4 shows a screen shot illustrating the debugger window, with sub-windows labelled. The above serves as an introduction to application development for the MC68HC12; for a more comprehensive introduction the reader is referred to [2].

5.2.2 MC68HC912B32 application code.

Here a comprehensive description of the operation of the microcontroller program used to implement both linear and non-linear algorithms is made. The main program loop is treated firstly, followed by descriptions of each subroutine and interrupt service routine. Each description is made with the aid of a flow chart to illustrate the operations performed by the code.

Due to the nature of the subject material, some prior knowledge regarding assembly level programming of microprocessor systems has been assumed. Reference to Motorola data books [3], [4] for the MC68HC12 is advised to aid understanding. Code listings for the complete application program can be located in appendix B.

Main program

The main program consists essentially of a never ending loop, in which the status of various flags and port bits determines program flow and actions to be taken. In addition the main program was responsible for calculation of the look-up tables associated with both linear and non-linear algorithms. Figure 5.2.5 illustrates the operations performed by the main program.

On execution of the main program the stack pointer was initialised to $0BFF, which is the last memory location of the processors internal RAM. After the stack has been initialised the processor resources are configured to suit the present application by executing the initialisation subroutine. To allow reception of data from the SCALA electronics host PC, the interrupt mask of the processor was cleared, thereby enabling the processor to service interrupt requests.

Port P bit 2 was interrogated to ascertain the status of the enable signal, which was active during STS. If the enable signal was active an STS measurement was in progress and no further action was taken until the signal became inactive.

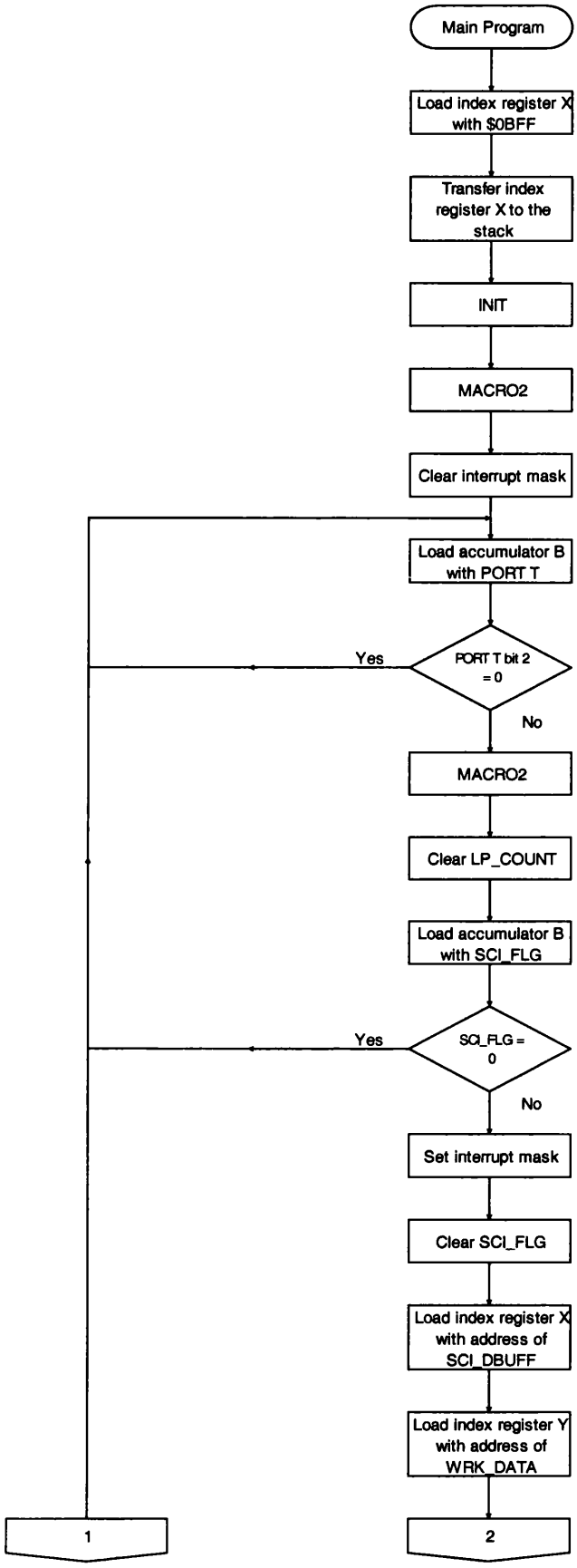


Figure 5.2.5: A flow chart illustrating the operations performed by the main program.

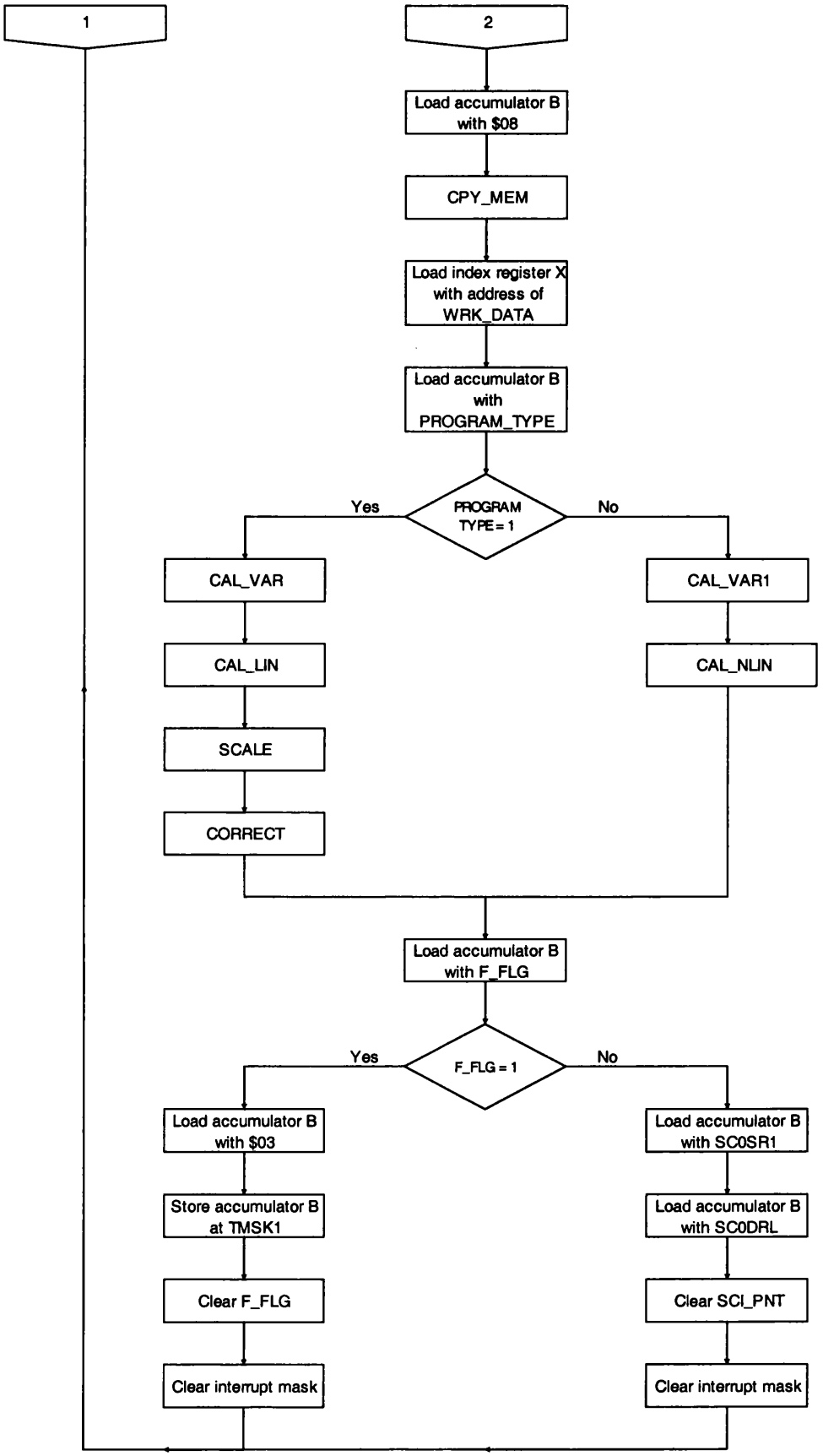


Figure 5.2.5: A flow chart illustrating the operations performed by the main program.

When the signal was inactive the output of the DAC module was cleared to 0v and the SCI flag, SCI_FLG interrogated. If SCI_FLG was clear a new data set was not available and the code returned to check the enable signal. However, if SCI_FLG was set, a new data set had been successfully received from the SCALA electronics host PC. At this point the interrupt mask was set to prevent service of any interrupts which occurred during ensuing calculations and the new data set was transferred from the SCI data buffer, SCI_DBUFF to the working data table, WRK_DATA. Accumulator B was loaded with PROGRAM_TYPE from the working data table. If the PROGRAM_TYPE = 1 the code relating to linear algorithm was executed, calculating a new look-up table by sequentially executing the subroutines related to the linear program. If the PROGRAM_TYPE = 2 the code relating to the non-linear algorithm was executed, calculating the new look-up table by executing subroutines related to the non-linear algorithm.

Accumulator B was loaded with the first run flag F_FLG to ascertain if the program was being executed for the first time following reset. If F_FLG was set bits 0 and 1 in the timer mask register TMSK1 were set to allow timer channels 0 and 1 to cause interrupts. If F_FLG was clear any pending interrupts from the SCI were cleared and SCI_PNT cleared. In both cases the interrupt mask was cleared before returning to check the enable signal.

Initialisation routine

The initialisation subroutine 'INIT' was responsible for configuring processor resources to suit the present application prior to entry into the main program loop. Figure 5.2.6 illustrates the actions performed by the initialisation subroutine.

On execution the subroutine configures processor resources to suit the requirements of the present system. Firstly, memory resources used by the program were cleared to $00 to prevent erroneous data from affecting the operation of the program. Secondly the ports utilised by the system were configured: ports A and B were configured for output only operation by writing $FF to data direction registers associated with the ports, DDRA and DDRB respectively; port P bits 0, 1 and 2 were configured as output only by writing $07 to DDRP.

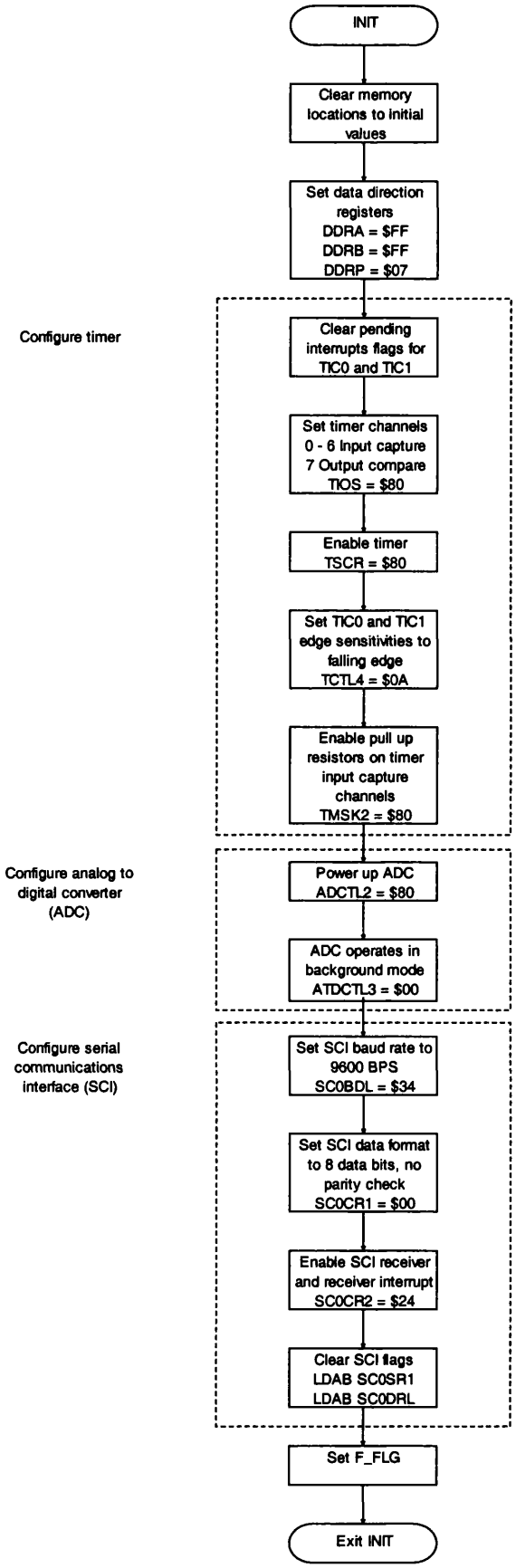


Figure 5.2.6: A flow chart illustrating the actions performed by the INIT sub-routine.

Thirdly, channels 0 to 6 of the timer module were configured for input capture operation and the remaining channel configured for output compare. Of the six channels configured as input capture only channels 0 and 1 were used. Channels 0 and 1 were configured to cause hardware interrupts on the falling edge of a TTL compatible signal presented to their respective physical pins, port T bit 0 and bit 1. The interrupt service routines associated with channels 0 and 1 are 'TIC0_INT' and 'TIC1_INT' respectively. Timer channel 7 was utilised by the general purpose delay sub-routine to facilitate timing of accurate delays. Fourthly, the on chip analog to digital converter was powered up by setting bit 7 of ADCTL2. Fifthly, the serial communications interface was configured to facilitate serial communication with a host PC. The SCI was configured to operate at 9600 baud rate with 8 bit data format and no parity check. SC0CR2 was configured to allow the receiver section of the SCI to cause an interrupt on the successful reception of a byte of data. When an interrupt occurred the associated service routine SCI_INT was executed. Finally the subroutine sets the first run flag F_FLG, indicating to the main routine that this was the first time the program has been executed after reset.

Calculation of linear variables

Calculation of mathematical constants STEP_SIZE and K used during generation of the linear algorithm look-up table was performed by the CAL_VAR sub-routine. Figure 5.2.7 illustrates the operations performed by the subroutine during calculation of STEP_SIZE and K.

On execution of the subroutine the DATA_POINTS value stored in the working data table WRK_DATA was transferred to accumulator B and $01 subtracted. Accumulator D which is a concatenation of accumulators A and B was divided by $02 and the quotient returned in index register X. Accumulator D was loaded with $0CCC, which represents 0.1 nm tip movement and divided by the contents of index register X. The quotient which was returned in index register X was stored at STEP_SIZE.

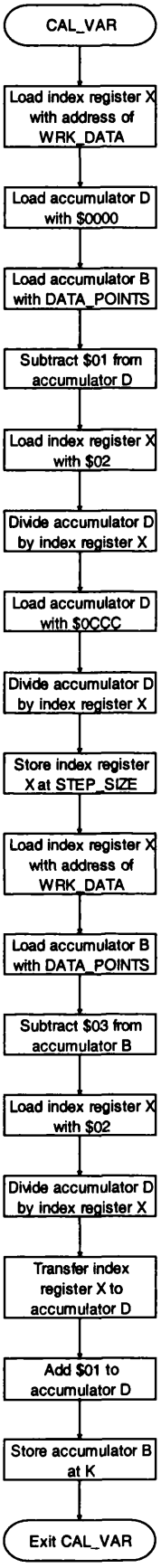


Figure 5.2.7: A flow chart illustrating the operations performed by the linear program variable calculation subroutine.

To calculate K, accumulator B was loaded with the DATA_POINTS value from WRK_DATA and $03 subtracted. Accumulator D was divided by $02 and the quotient returned in index register X transferred to accumulator D and $01 added. The result of the calculation which is always < 255 was stored at K.

Calculation of linear program look-up table

The calculation of the look-up table associated the linear program was performed by the 'CAL_LIN' subroutine. To simplify calculations performed, the subroutine calculates a look-up table which was normalised to 0.1nm tip displacement. To achieve greater tip displacement from the regulated height the scaling subroutine multiplies each entry within the table by a scaling factor. The number of entries in the look-up table was defined by the number of spectroscopy data points. Figure 5.2.8 shows the operations performed by the subroutine during calculation of the linear program look-up table.

On execution of the subroutine the DATA_POINTS value located in the working data table was transferred to accumulator B and $02 subtracted, the result was stored in the loop counter LP_COUNT, thus defining the number of entries within the table minus the first and last, which were always $0000. Index register X was loaded with the address of the first entry in the look-up table, TAB_START. Accumulator D was loaded with $0000 and stored at the address pointed to by index register X and at the temporary memory location TEMP. Index register X was incremented to point to the next table entry. Accumulator B was loaded with LP_COUNT and tested; the result determining the action to be taken. If LP_COUNT = 0, $0000 was stored as the last table entry and the routine terminated. If LP_COUNT <= K, STEP_SIZE was added to TEMP and the result stored at the address specified by index register X, otherwise STEP_SIZE was subtracted from TEMP and the result stored at the address specified by index register X. LP_COUNT was decremented and index register X pointed to the next table entry and the process repeated until LP_COUNT = 0.

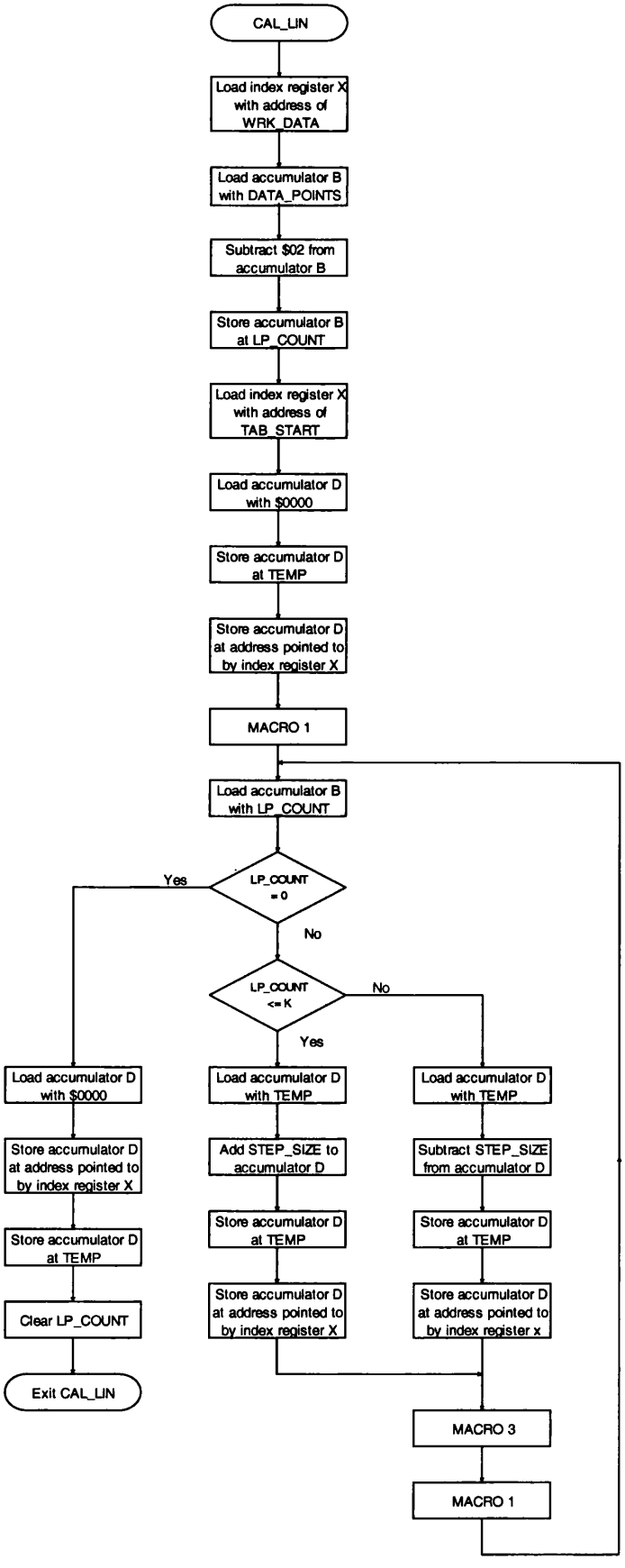


Figure 5.2.8: A flow chart illustrating the operations performed by the linear program look-up table calculation subroutine.

Scaling routine

The scaling subroutine 'SCALE' was responsible for scaling the normalised look-up table generated by the linear program sub-routine. Each entry in the look-up table was multiplied by a scaling factor representing the peak tip displacement from the regulated height. Figure 5.2.9 illustrates the operations performed by the scaling routine.

On execution of the sub-routine the DATA_POINTS value located in the working data table was copied to LP_COUNT, which defined the number of entries in the look-up table. Index register X was loaded with the address of the first entry in the look-up table, TAB_START. If the value in LP_COUNT > 0, the table entry at the address specified by index register X was copied to accumulator D and multiplied by the scaling factor. The result was returned to the same table entry and the loop counter decremented. Index register X was incremented to specify the next table entry and the process repeated until LP_COUNT = 0, at which point the sub-routine terminated.

Correction routine

The correction routine 'CORRECT' was responsible for introducing an offset into the look-up table which accounts for the bipolar zero [5] of the DAC module. To accomplish this, each entry in the look-up table was subtracted from $8000. Figure 5.2.10 illustrates the operations performed by the correction routine.

On execution of the subroutine the DATA_POINTS value located in the working data table was copied to LP_COUNT, which defined the number of entries in the look-up table. Index register X was loaded with the address of the first entry in the look-up table, TAB_START. If the value in LP_COUNT > 0, accumulator D was loaded with the DAC module bipolar zero and the table entry to which index register X pointed was subtracted form accumulator D. The result was returned to the same table entry and the loop counter decremented. Index register X was incremented to specify the next table entry and the process repeated until LP_COUNT = 0, at which point the subroutine terminated.

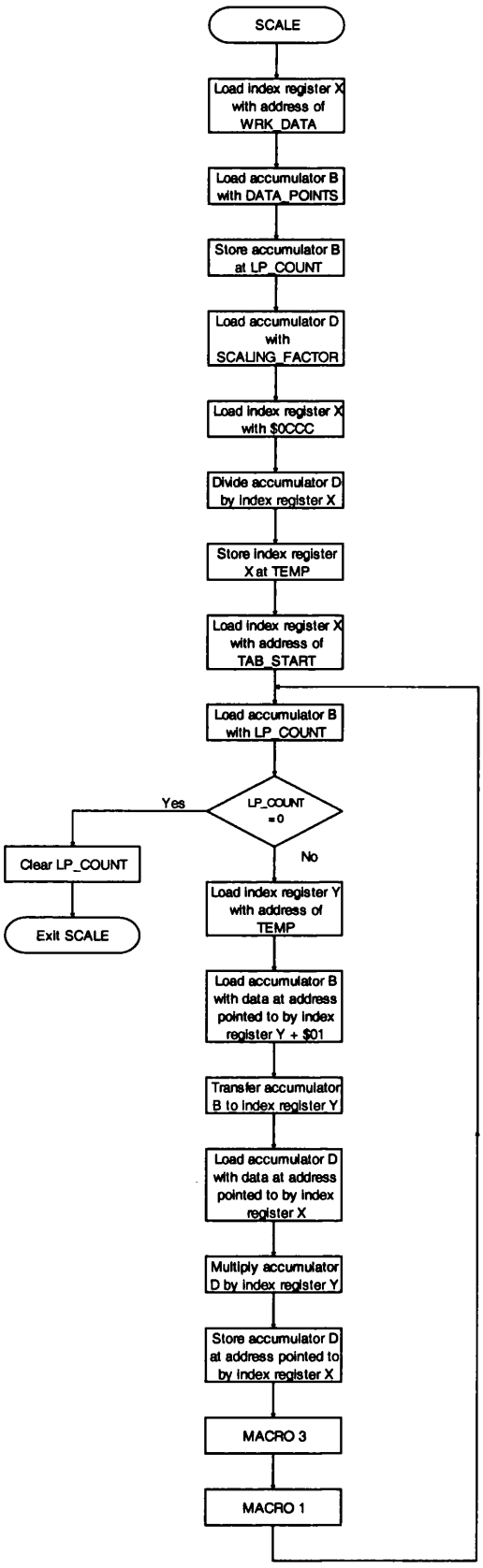


Figure 5.2.9: A flow chart illustrating the operations performed by the scaling sub-routine.

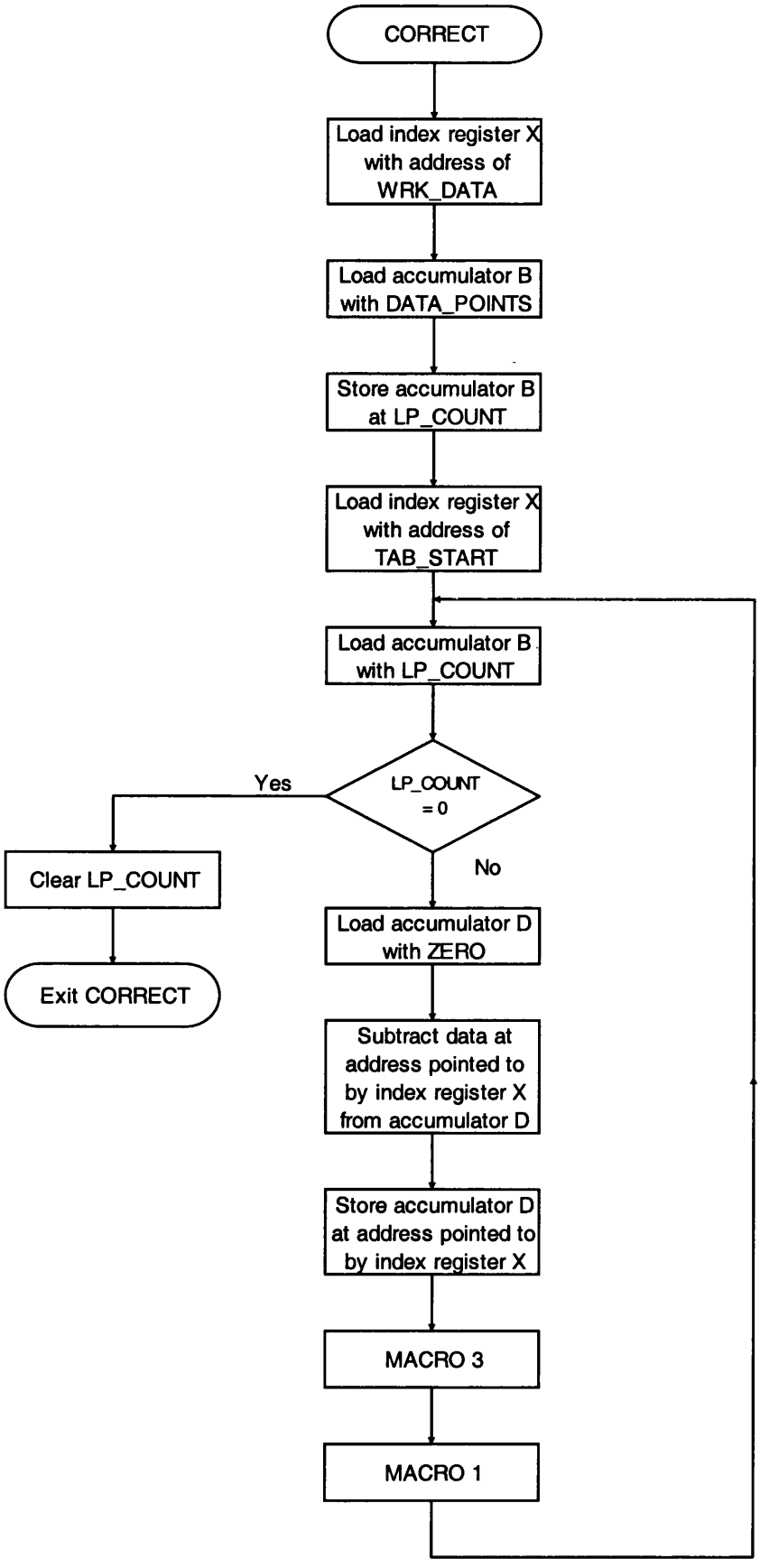


Figure 5.2.10: A flow chart illustrating the operations performed by the correction routine.

Calculation of non-linear variables

Calculation of the mathematical constant K1 used during the calculation of the non-linear look-up table was performed by the 'CAL_VAR1' subroutine. Figure 5.2.11 illustrates the operations performed by the subroutine during calculation of K1.

On execution of the subroutine the DATA_POINTS value was transferred from the working data table to accumulator B and $01 subtracted. Accumulator D was divided by index register X and the resulting quotient that was returned in index register X transferred to accumulator D. Since the result of the division was always < 255, only accumulator B which forms the lower byte of accumulator D was stored at K1.

Calculation of non-linear program lookup table

The calculation of the look-up table associated with the non-linear program was performed by the 'CAL_NLIN' subroutine. The subroutine was responsible for calculating a look-up table containing 256 consecutive data entries. Figure 5.2.12 illustrates the operations performed by the subroutine during calculation of the look-up table associated with the non-linear program.

On execution of the subroutine, accumulator D was loaded with MAX_STEP_SIZE from the working data table; index register X was loaded with $FF and used to divide accumulator D and the result of the division stored in TEMP. Index register X was loaded with the address of the first entry in the look-up table TAB_START, and accumulator D with $0000. Since the first entry in the table is $0000, accumulator D was stored at the address index register X pointed to and then the address in X pointed to the next table entry. Accumulator B was loaded with $FF and stored in LP_COUNT to define the length of the look-up table. Accumulator B was loaded with LP_COUNT and tested. If LP_COUNT = 0, the routine terminated clearing LP_COUNT and TEMP, otherwise accumulator D was loaded with the previous table entry, TEMP added and the result stored at the table entry index register X pointed to. LP_COUNT was decremented and index register X pointed to the next table entry and the process repeated until LP_COUNT = 0, at which point the routine terminated.

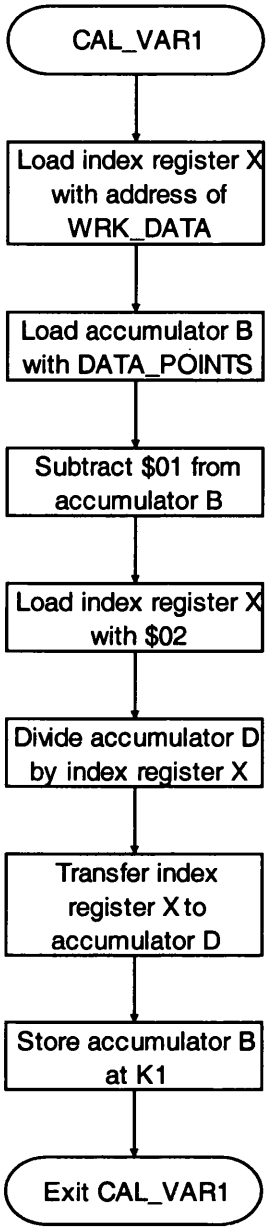


Figure 5.2.11: A flow chart illustrating the operations performed by the subroutine responsible for calculating K1.

General purpose delay

The general purpose delay subroutine 'GP_DLY' utilises timer output compare 7 (TOC7) to provide delays ranging from us to ms in duration. The desired delay was stored in DLY_VAL prior to execution of the sub-routine. Figure 5.2.13 illustrates the actions performed by general purpose delay subroutine. For details regarding the operation of the timer module of the MC68HC12 refer to [6].

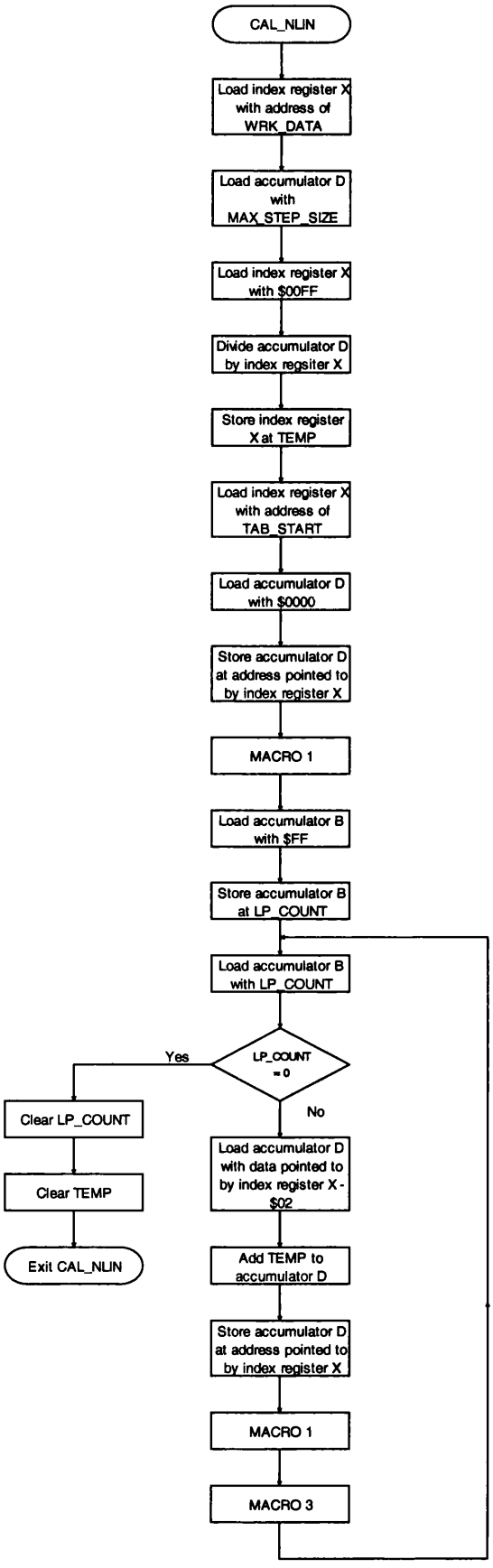


Figure 5.2.12: A flow chart illustrating the operations performed by the subroutine responsible for calculating the look up table associated with the non-linear program.

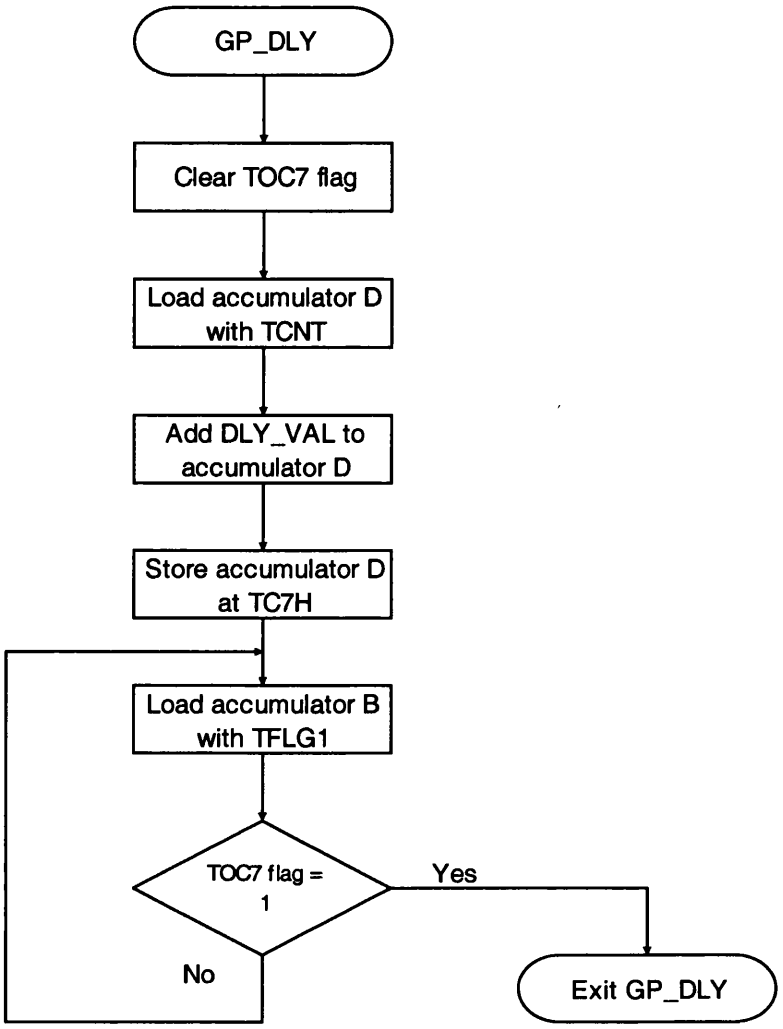


Figure 5.2.13: A flow chart illustrating the operations performed by the general purpose delay subroutine.

On execution of the subroutine, the flag relating to an interrupt generated by timer channel 7 in the timer flag register TFLG1 was cleared. The current value of the timer count register TCNT was transferred to accumulator D and the value stored in DLY_VAL added to it. Accumulator D was transferred to timer channel 7 input capture / output compare register TC7, setting up the delay period. Accumulator B was loaded with TFLG1 and timer channel 7 flag examined. If the flag was set, the value in TCNT was equal to the value in TC7 and the delay period had elapsed, at which point the routine terminated; otherwise the flag was repeatedly checked until the above condition was true.

Clear memory

The clear memory sub-routine 'CLR_MEM' was responsible for clearing n consecutive memory locations to $00, and was primarily used during processor initialisation. Figure 5.2.14 illustrates the operations performed by the clear memory subroutine.

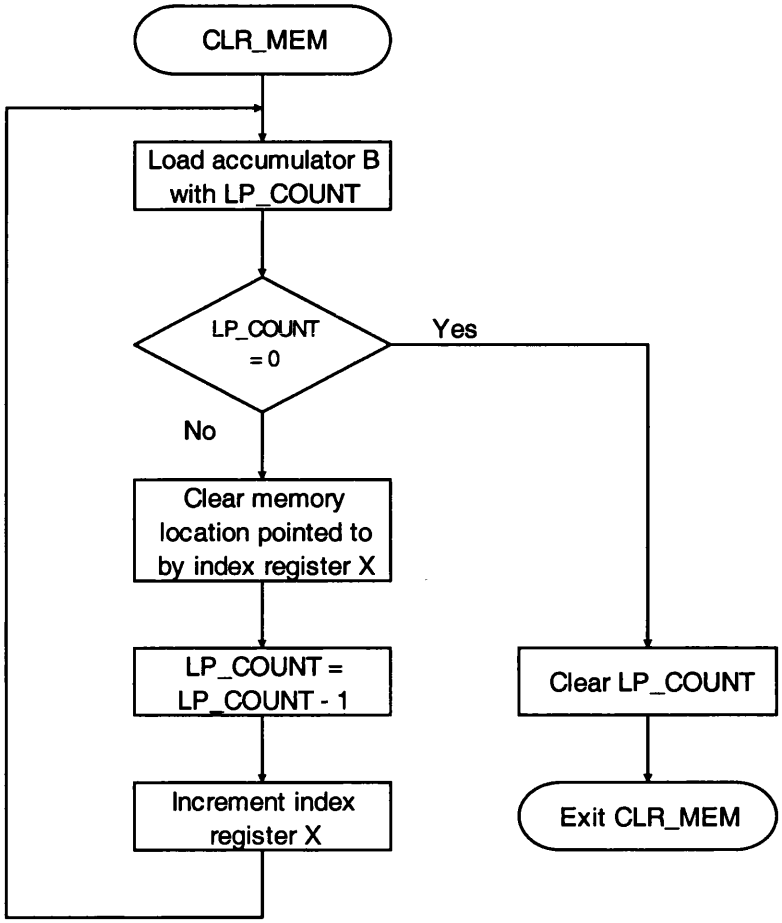


Figure 5.2.14: A flow chart illustrating the operations performed by the clear memory sub-routine.

LP_COUNT and index register X are loaded with n and the start address of memory to be cleared respectively, prior to execution of the subroutine. On execution, LP_COUNT was transferred to accumulator B and tested. If LP_COUNT was > 0, the memory location that index register X pointed to was cleared. LP_COUNT was decremented and index register X incremented and the process repeated until LP_COUNT = 0, at which point the subroutine terminated.

Copy memory

The copy memory subroutine 'CPY_MEM' was responsible for copying n consecutive memory locations whose start address was specified by index register X, to n consecutive locations whose start address was specified by index register Y. Figure 5.2.15 illustrates the operations performed by the copy memory subroutine.

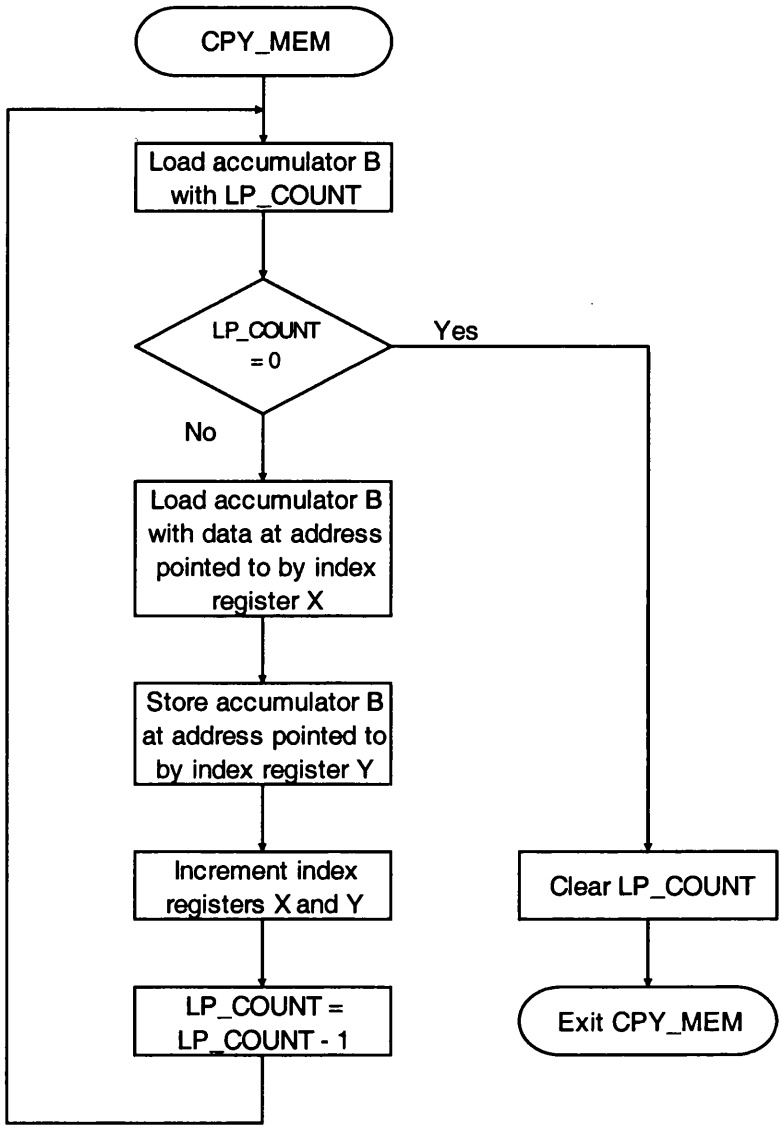


Figure 5.2.15: A flow chart illustrating the operations performed by the copy memory subroutine.

LP_COUNT, index register X and index register Y are loaded with n, the start address of source memory and the start address of the target memory respectively, prior to execution of the subroutine. On execution LP_COUNT was transferred to

accumulator B and tested. If LP_COUNT > 0, the memory location that index register X pointed to was copied to the memory location index register Y pointed to. LP_COUNT was decremented and index registers X and Y incremented and the process repeated until LP_COUNT = 0, at which point the routine terminated.

Write DAC

The write to DAC subroutine 'WRT_DAC' was responsible for writing a data word stored in Z_OUT to the DAC module. Z_OUT was transferred to accumulator D prior to execution of the subroutine. Microcontroller ports A and B are combined to provide a single 16bit port, which connects directly to D0 – D15 of the DAC module. Bits 0, 1 and 2 of port P provide control signals for the DAC module. Figure 5.2.16 illustrates the operations performed by the write to DAC subroutine.

On execution of the subroutine ports A and B are cleared. The high and low bytes of the data word were transferred from accumulators A and B to ports A and B respectively (accumulator D is formed from accumulators A and B). To write the contents of ports A and B to the converter, control codes are written sequentially to port P, latching a data word into the DAC module. IDLE ($06) was transferred to port P at the beginning and completion of a write cycle, ensuring the converter was in an idle state prior to and following a transfer. The converters internal data latches were loaded by transferring L1 ($00) followed by L2 ($06) to port P. Comprehensive details regarding the operation of the DAC module are given in 4.4 and not reproduced here.

The serial communications interface interrupt service routine, 'SCI_INT'.

When an interrupt was generated by the serial communications interface [7] (SCI) the associated interrupt service routine 'SCI_INT' was executed. This service routine was responsible for collecting a complete data set from the PC communications interface, and passing it to the main program on completion of the transfer. Table 5.2.1 shows the order in which the data set is received and the SCI_DBUFF locations and SCI_PNT values associated with each data type.

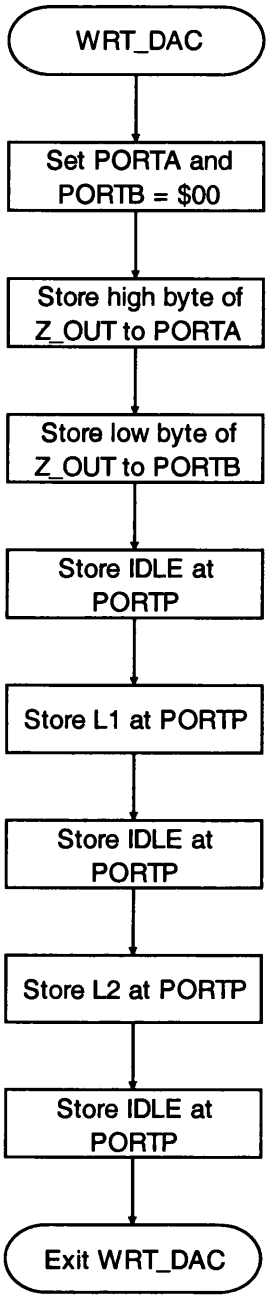


Figure 5.2.16: A flow chart illustrating the operations performed by the write to DAC subroutine.

Discussion of the SCI interrupt service routine proceeds with reference to the flow chart detailing the routine, depicted in figure 5.2.17. Register SC0SR1 which is associated with the SCI was interrogated to ascertain the type of interrupt generated by the SCI. If an interrupt was generated by reception of a data byte, the receiver data flag 'RDRF' in SC0SR1 was set and the routine proceeds, otherwise the routine terminated. The SCI flag SCI_FLG was interrogated to ascertain if the SCI data buffer

SCI_DBUFF had data waiting to be serviced by the main program. If SCI_FLG was set the service routine terminated; otherwise it proceeds.

Type	SCI_DBUFF value	SCI_PNT value	Comment
Program type	Location 0	$00	Indicates program type: 1 is linear, 2 is non-linear
Data points	Location 1	$01	Used by both linear and non-linear programs
Scaling factor	Locations 2 and 3	$02 and $03	Used by the linear program
Maximum step size	Locations 4 and 5	$04 and $05	Used by the non-linear program
Z-Limit	Locations 6 and 7	$06 and $07	Used by the non-linear program

N.B. a numerical value with a $ prescript indicates hexadecimal format.

Table 5.2.1: Data reception order and associated SCI_DBUFF and SCI_PNT values.

If the tests performed on RDRF and SCI_FLG are true the service routine proceeds to pick up the data waiting in the SCI receiver data register SC0DRL, and then transfers it to one of eight locations in SCI_DBUFF. To accomplish this, the first address of SCI_DBUFF was loaded into accumulator D and a pointer value stored in SCI_PNT added to the first address. SCI_PNT pointed to the next available location in SCI_DBUFF and can take on values $00 – $07. When SCI_PNT = $07, SCI_FLG was set to indicate SCI_DBUFF had data waiting and the routine terminated. If SCI_PNT was < $07 its value was incremented to point to the next available location in SCI_DBUFF and the routine terminated without SCI_FLG being set.

Timer interrupt 0

Timer channel 0 was configured for input capture operation and to cause a hardware interrupt on the falling edge of a TTL compatible signal presented to port T bit 0. When an interrupt occurred the associated service routine TIC0_INT was executed. The service routine provided a mechanism whereby the Z axes tip displacement for both linear and non-linear algorithms could be applied.

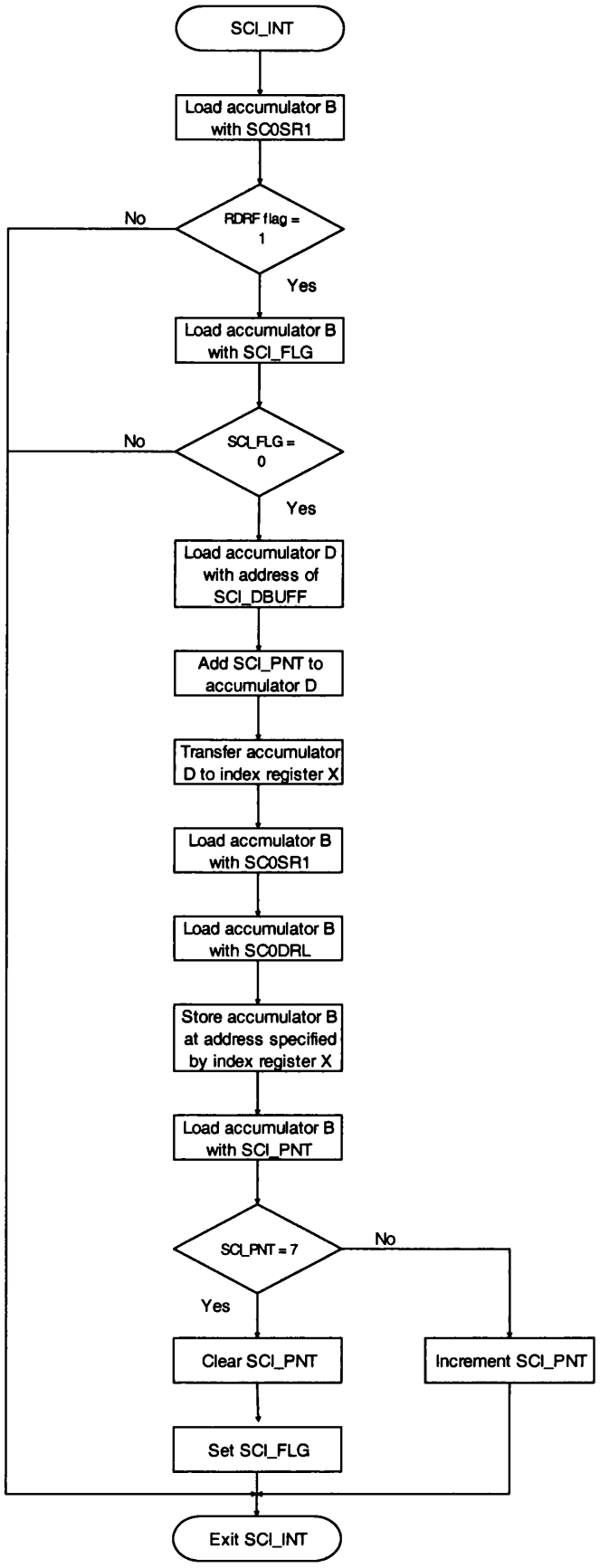


Figure 5.2.17: A flow chart illustrating the operations performed by the SCI interrupt service routine.

The service routine is considered in three distinct sections: a portion for deciding if a genuine interrupt has occurred and the program type being executed, a portion responsible for implementing code relating to the linear algorithm and a portion responsible for implementing code relating to the non-linear algorithm.

On entry to the portion of code responsible for determining the validity of the interrupt, the flag associated with timer channel 0 was cleared to prevent re-service of the interrupt on termination of the service routine. To ascertain if a genuine interrupt had occurred, port T bit 2 is interrogated. If bit 2 was clear the routine proceeds, otherwise the routine terminated with no further action. The program flag, P_FLG was examined to ascertain the type of program being executed. If P_FLG = 1, code pertaining to the linear algorithm was executed, if however P_FLG = 2, code pertaining to the non-liner algorithm was executed, otherwise the routine terminated with no further action. Figure 5.2.18 illustrates the operations performed by the portion of code responsible for interrupt validity verification and program type checks.

When P_FLG = 1 the portion of code relating to the linear algorithm was executed and the table entry whose address was specified by the table pointer TPT was written to the DAC module. Successive interrupts result in the look-up table related to the linear algorithm being sequentially written to the DAC module. The address specified by TPT was pointed to the next valid table entry before the service routine terminated. Figure 5.2.19 illustrates the operations performed by the code responsible for the implementing linear program.

Accumulator B was loaded with LP_COUNT and tested. If LP_COUNT = 0, accumulator B was loaded with the DATA_POINTS value from the working data table, decremented and stored at LP_COUNT. In addition accumulator D was loaded with the address of the first entry of the look-up table related to the linear algorithm and stored at TPT. If LP_COUNT > 0 accumulator B was decremented and stored at LP_COUNT. Accumulator D was loaded with the table entry whose address was specified by TPT and then written to the DAC module. The address within TPT was pointed to the next table entry and the routine terminated with no further action.

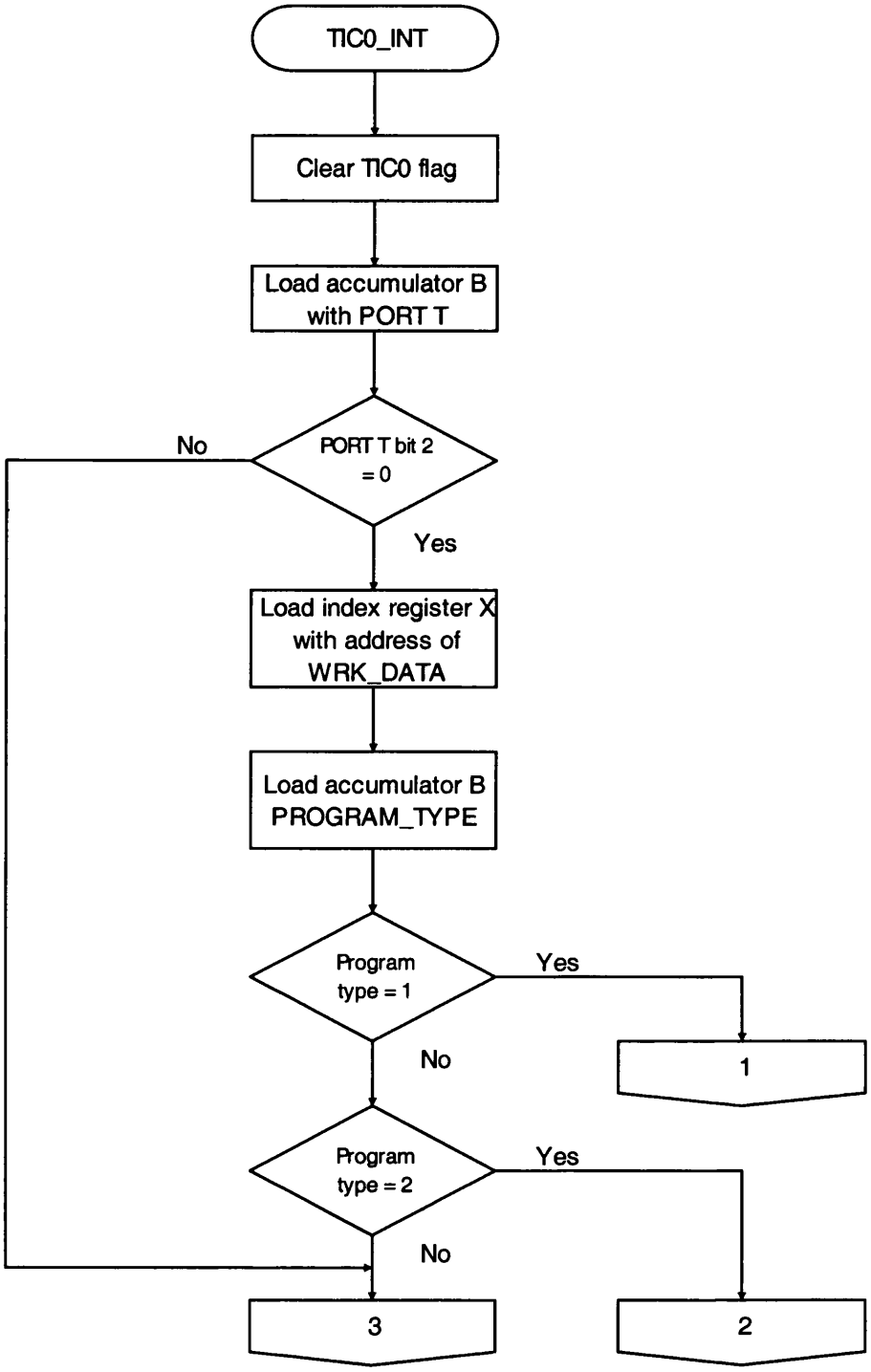


Figure 5.2.18: A flow chart illustrating the operations performed by the portion of code responsible for interrupt validity and program type checks.

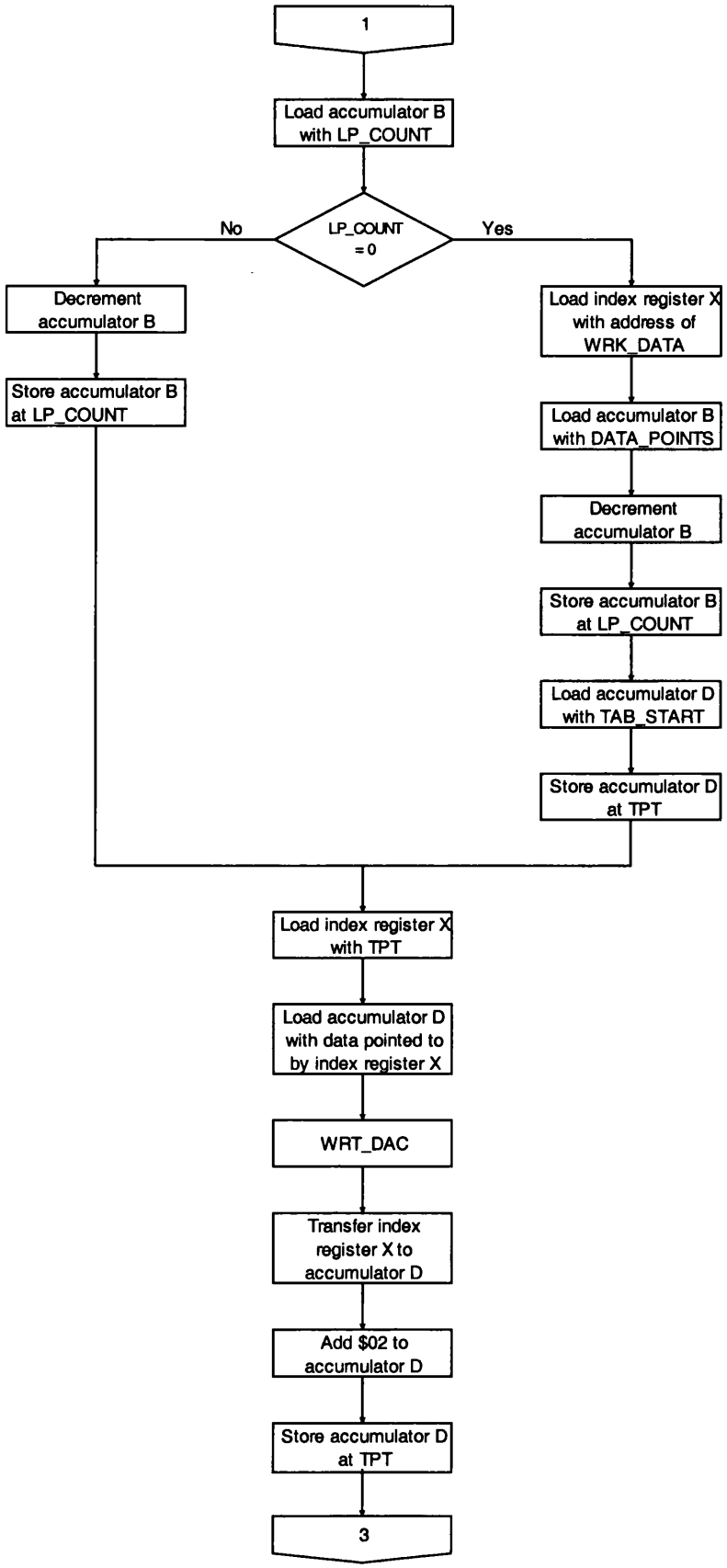


Figure 5.2.19: A flow chart illustrating the operations performed by the code responsible for implementing the linear program.

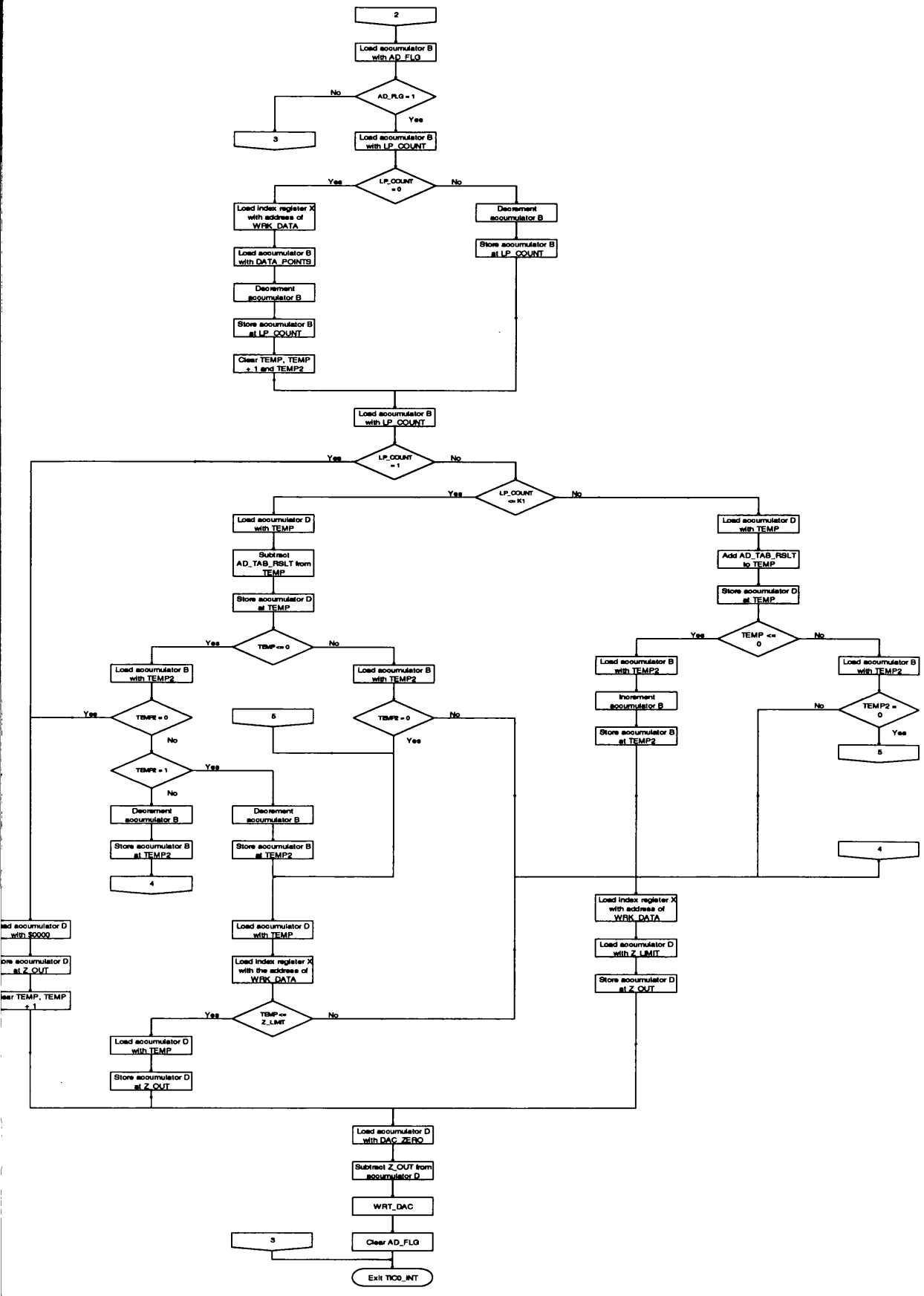


Figure 5.2.20: A flow chart illustrating the operation of the section of code responsible for the non-linear program.

When P_FLG = 2 the portion of code responsible for implementing the non-linear algorithm section of the interrupt service routine was executed. Figure 5.2.20 illustrates the operations performed by the portion of code relating to the non-linear algorithm. Due to the complexity of the non-linear code portion a terse description of the operation is undertaken in which the detail level is reduced to aid lucidity.

To ascertain if new data was waiting in AD_TAB_RSLT, AD_FLG was interrogated. If AD_FLG was set, new data was available and the routine proceeded, otherwise the routine terminated with no further action. LP_COUNT was examined to ascertain the position which had been reached during measurement of a spectra. If LP_COUNT = 0, the last point in a measurement spectra had been reached and the LP_COUNT was reset, taking on the DATA_POINT value – 1; otherwise the loop counter was decremented.

The remaining portion of code was responsible for calculating an offset to be applied to a running accumulator stored at the temporary memory location TEMP, comparing the value within TEMP to predefined limits and taking appropriate action if the limits were violated. The value to be applied was temporally stored within Z_OUT; corrected for the bipolar zero of the DAC module and written to the DAC module.

Timer interrupt 1

Timer channel 1 was configured for input capture, and to causes a hardware interrupt on the falling edge of a TTL compatible signal presented to port T bit 1. When an interrupt occurred the associated service routine, TIC1_INT was executed. The service routine was responsible for sampling an analog voltage and manipulating its digital equivalent to generate a table address within the non-linear algorithm look-up table from which a result to be used by TIC0_INT was obtained. Figure 5.2.21 illustrates the operations performed by timer channel 1 interrupt service routine.

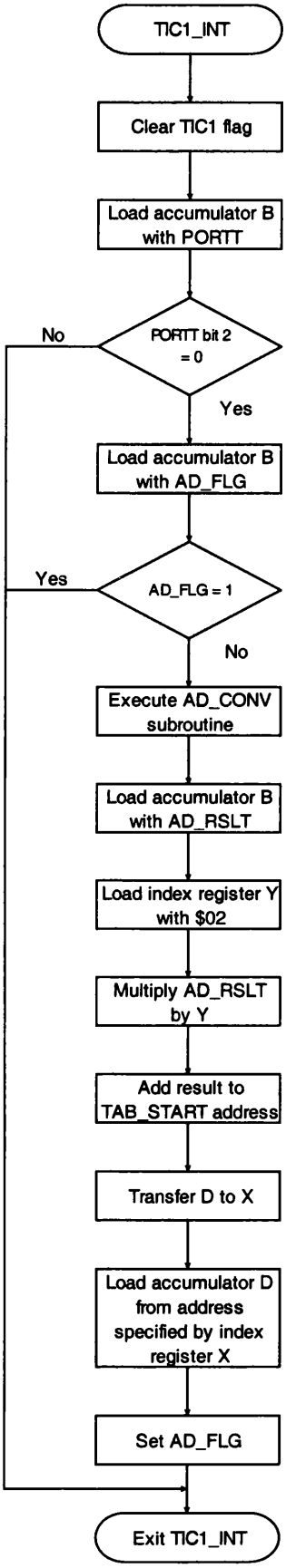


Figure 5.2.21: A flow chart illustrating the operations performed by timer channel 1 interrupt service routine.

On entry to the service routine, the flag associated with timer channel 1 in TFLG1 was cleared to prevent re-service of the interrupt on termination of the service routine. The state of port T bit 2 was checked to ensure a genuine interrupt had occurred. If bit 2 was clear the routine proceeded, otherwise the routine terminated with no further action. To ensure the result from the previous execution of the service routine had been used AD_FLG was interrogated. If AD_FLG was not set the service routine proceeded, otherwise the routine terminated with no further action.

On successful completion of the above tests the routine proceeded to sample the analog voltage presented to port AD bit 0 by executing ADCONV. The result returned, AD_RSLT was multiplied by $02 and added to TAB_START to generate the address within the non-liner look-up table from which AD_TAB_RSLT was loaded. AD_FLG was set to indicate the presence of a new result in AD_TAB_RSLT and the routine terminated.

5.3 The graphical user interface operation and development tools employed.

5.3.1 Borland Delphi.

Borland Delphi is a windows based event driven graphical programming language, which encompasses and is based around the Pascal programming language. Figure 5.3.1 shows a screen shot of the Delphi development environment. A brief description of application development ensues with reference to figure 5.3.1.

To create a new application components are chosen from the component palette and positioned on a blank form. The properties and events associated with each component are then tailored to the application by selecting a particular property or event from the object inspector and configuring it. Properties describe the appearance and function of a component, whereas events generally but not exclusively occur when the user interacts with component at run time. Each form has a code unit associated with it containing user developed Pascal code relating to the form and its associated components. Procedures relating to specific components or componential

events are added to the unit as required. Code relating to a specific event is executed when the event occurs at run time. At completion of the application any coding errors are flagged during compilation and may be corrected. If compilation is successful an executable file is generated and the application launched. The above serves as an introduction to Delphi application development, for a more comprehensive introduction and guide to Delphi and Pascal the reader is referred to [8], [9].

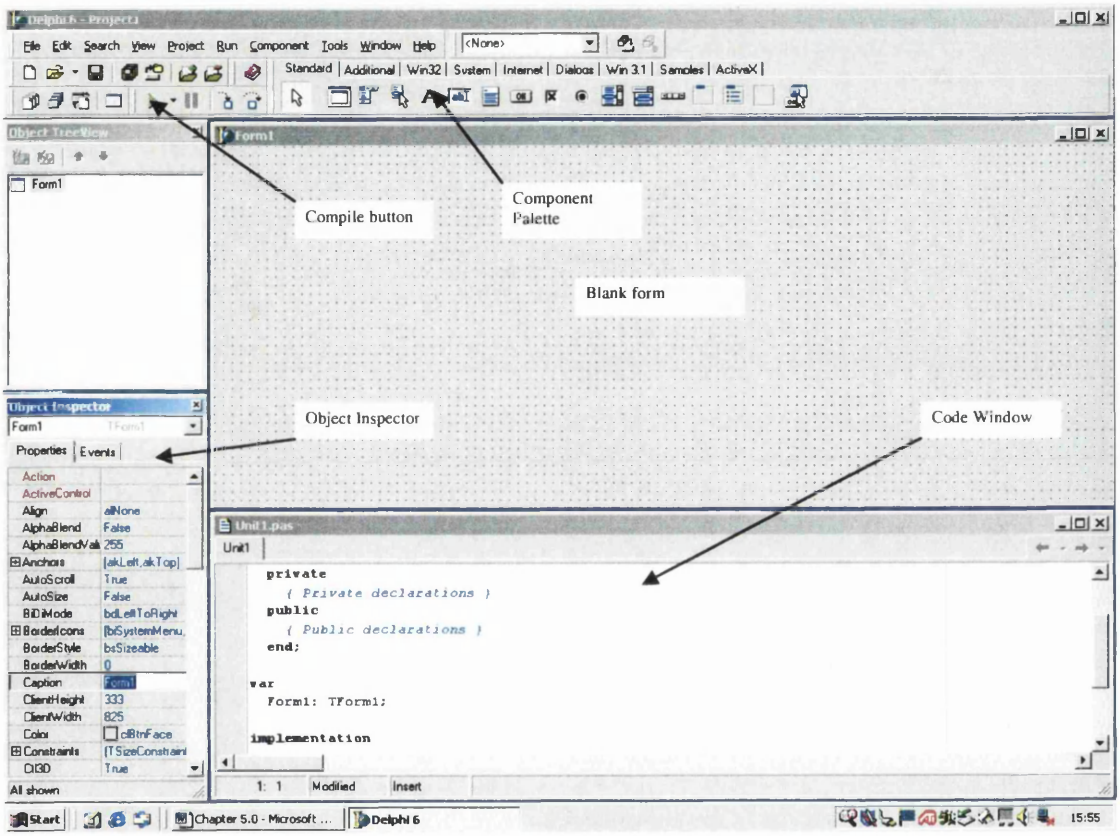


Figure 5.3.1: Screen shot of the Delphi development environment.

5.3.2 The graphical user interface.

Here a description of the variable tip-sample separation graphical user interface is undertaken, with emphasis placed on its functionality as opposed to a detailed analysis of the interface coding. Following launch of the variable tip-sample separation application the user is presented with the main dialog, figure 5.3.2. The main dialog comprises a menu with four catagories: 'File', 'Program', 'Communication' and 'Help'. Additionally, a panel at the bottom of the main dialog displays error messages specifically related to errors encountered whilst opening program parameter files.

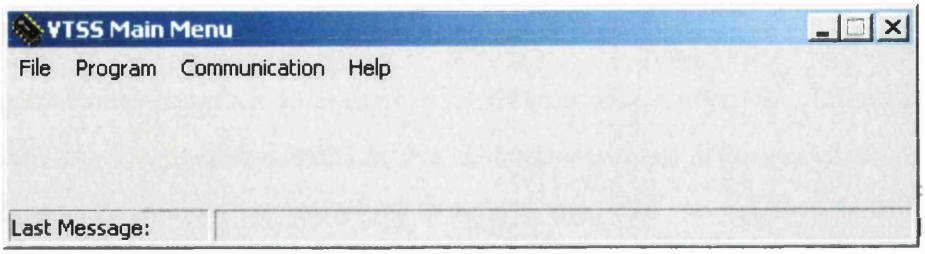


Figure 5.3.2: Main dialog.

Selection of the 'File' category produces a drop menu, figure 5.3.3 with: 'Save', 'Open' and 'Exit' options.

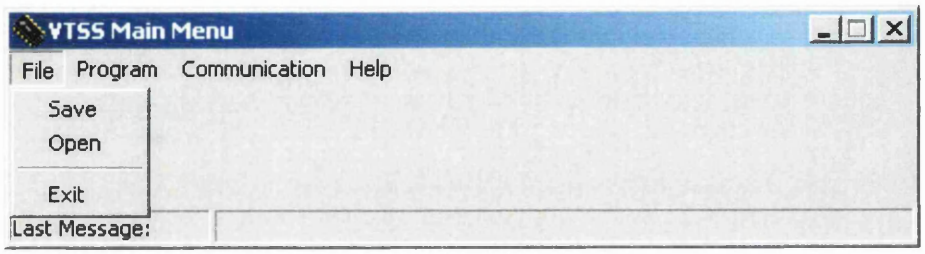


Figure 5.3.3: File menu in the main dialog.

Choice of the 'Save' option produces a 'Save As' dialog, figure 5.3.4 allowing program parameters to be saved to file. The save option is available only after selection of either 'Liner' or 'Non-linear' options from the program menu. Program parameter files contain parameters relating to linear and non-linear programs, communications and environmental settings. Figure 5.3.5 shows the contents of a program parameters file; Example.dat

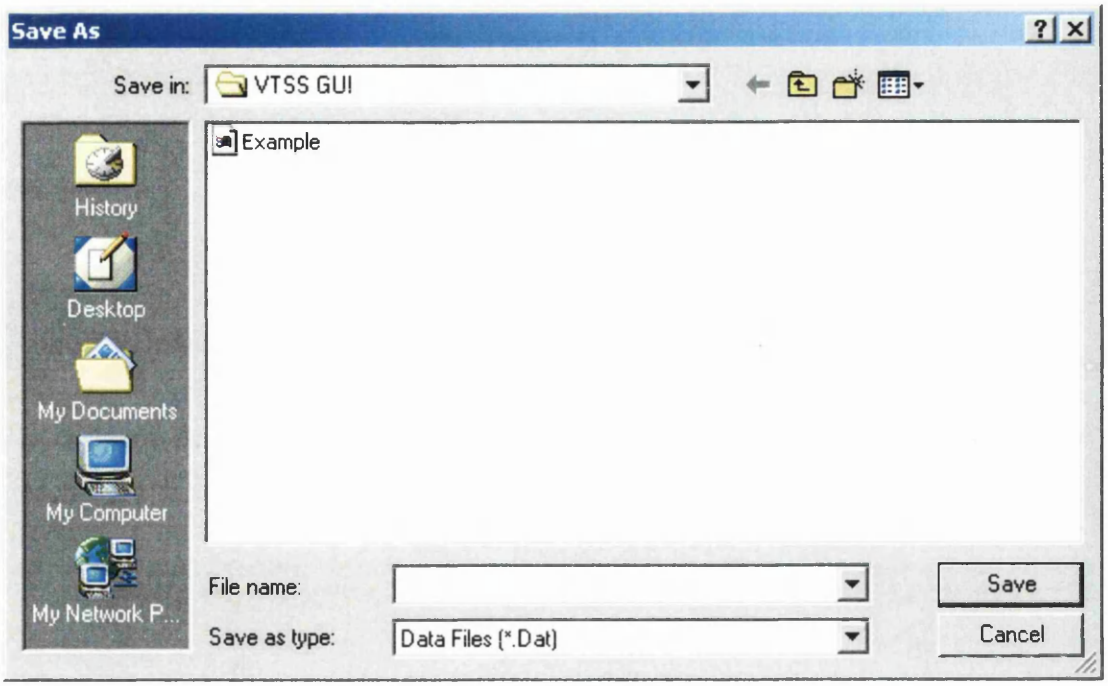


Figure 5.3.4: Save As Dialog.

FileName: C:\Documents and Settings\James\My Documents\VARAIBLE TIP-SAMPLE SEPARATION GUI\Example.dat

Date/Time: 21/10/2002 18:05:52

Data Section

Program Type:
1
Data Points:
1
Scaling Factor:
0
Maximum Step Size:
0
Z_Limit:
0

Enviromental Parameters

Serial Port DLG Open(1)/Close(0):
0
Comm Port:
1
Baud Rate:
6
Parity:
0
Flow Control:
0
Data Bits:
4
Stop Bits:
0

Figure 5.3.5: Example program parameters file, Example.dat.

Choice of the 'Open' option produces an 'Open' dialog, figure 5.3.6 from which program parameter files may be loaded.

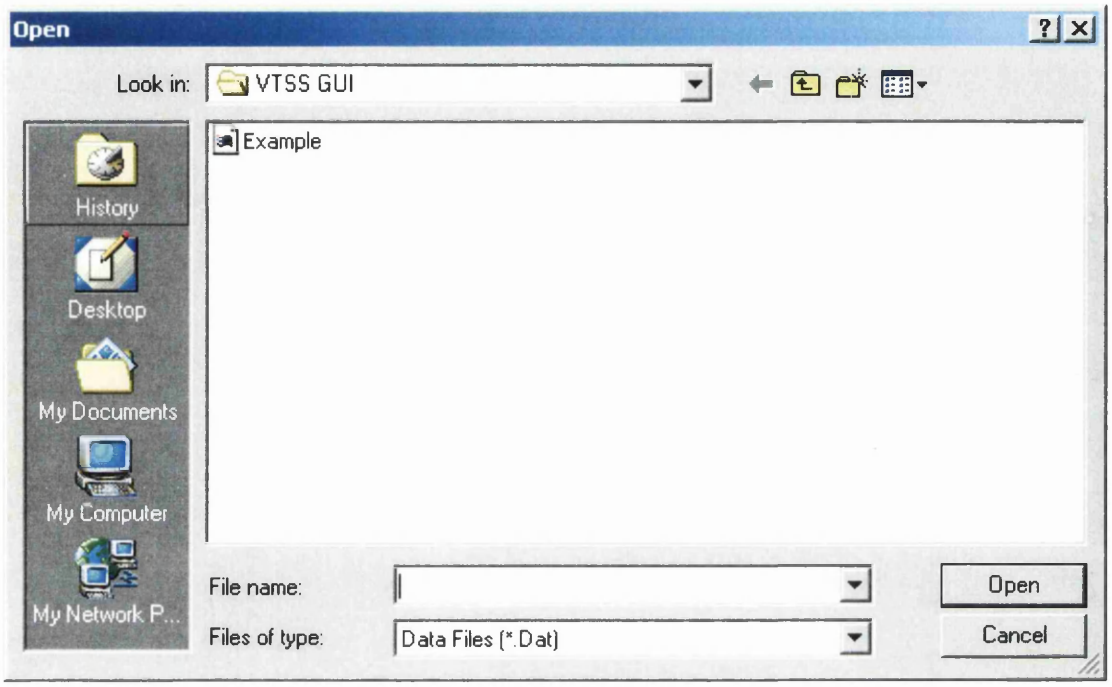


Figure 5.3.6: Open dialog.

Choice of the 'Exit' option produces a message dialog, figure 5.3.7 which functions as an exit confirmation. The same dialog is produced when the 'Close' border icon is selected. Program parameters not saved to file are lost on exit.

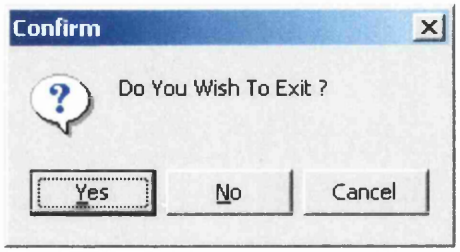


Figure 5.3.7: Exit conformation.

Selection of the 'Program' category produces a drop down menu, figure 5.3.8 with 'Linear' and 'Non-Linear' options. Discussion of both linear and non-linear program dialogs ensues.

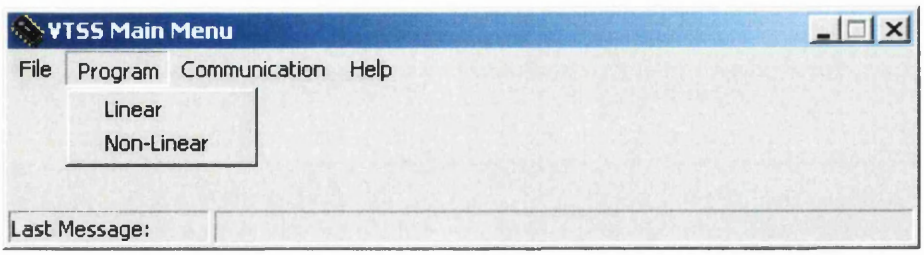


Figure 5.3.8: Program menu in the main dialog.

The 'Linear Program Parameters' dialog, figure 5.3.9, facilitates selection of parameters specific to the linear program, these being the number of data points and the scaling factor.

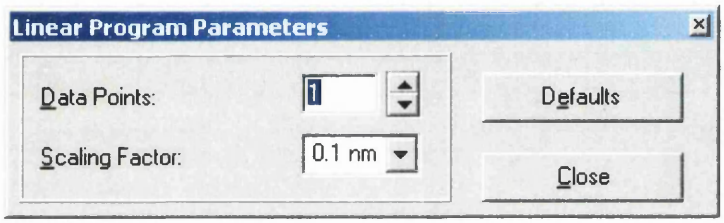


Figure 5.3.9: Linear Program Parameters dialog.

The value in the 'Data Points' field can take on odd integer values between 1 and 255, directly relating to the number of data points in a measured spectra. Since the linear tip displacement contour is symmetrical, a single data point is required at the apex of the contour with equal numbers of data points to each side of the apex, hence the necessity for an odd number of data points. Scaling factor values in the range 0.1 to 0.9nm in 0.1nm increments, are accommodated. Selection of the scaling factor determines the peak tip displacement from the regulated tip height achieved during spectra acquisition, which occurs at the data point ((n-1)/2) + 1, where n is equal to the number of data points in a measured spectra. Default values of 1 data point and 0.1nm scaling factor are selected by toggling the 'Defaults' button.

The 'Non-Linear Program Parameters' dialog, figure 5.3.10 facilitates selection of parameters specific to the non-linear program, these being: the number of data points, the maximum tip displacement from regulated tip height, 'Z Limit' and the maximum single step size, 'Maximum Step Size'.

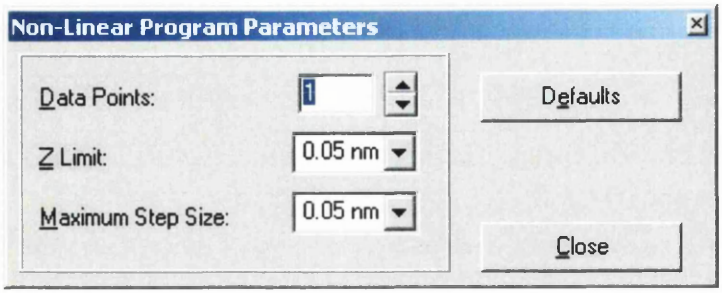


Figure 5.3.10: Non-Linear Program Parameters dialog.

Z-limit values in the range 0.05 to 0.95nm in 0.05nm increments are accommodated. Selection of the Z-limit determines the peak tip displacement which can occur; unlike the linear program the Z-limit may be reached at any point during spectra acquisition. Maximum step sizes in the range 0.05 to 0.2nm in 0.05nm increments are accommodated, relating to the maximum tip movement in any single step. Default values of 1 data point, 0.05nm Z limit and 0.05nm maximum step size are selected by toggling the 'Defaults' button. A detailed explanation of the parameters used in both linear and non-linear dialogs is provided in section 5.1, which details the algorithms employed for both linear and non-linear program types.

An alternative method of selecting program parameters is to create or modify an existing program parameters file, such as figure 5.3.5. It is preferable to modify an existing file since parameters must be entered on specific lines of the file. The task of creating or modifying a program parameters file may be accomplished with a text editor capable of handling ASCII text format. Table 5.3.1 summarises valid data field values and their associated parameter values.

Data field name	Text file line number	Valid values	Program Parameter
Program Type	7	1,2	1 = Linear 2 = Non-Linear
Data Points	9	1,3..253,255	Data points value
Scaling Factor	11	0 – 8	0.1 - 0.9nm
Maximum Step Size	13	0 – 3	0.05 – 0.2nm
Z_Limit	15	0 – 19	0.05 – 0.95nm
Serial Port DLG Open (1)/Close(0)	20	0,1	Serial Port DLG window status 0 = Closed 1 = Open

Comm Port	22	0 – 7	COMM 1 – 8
Baud Rate	24	0 – 14	Baud Rate 110 to 256000. 9600
Parity	26	0 – 4	0 = None 1 = Odd 2 = Even 3 = Mark 4 = Space
Flow Control	28	0 – 3	0 = None 1 = XOn – Xoff 2 = RTS – CTS 3 = DSR – DTR
Data Bits	30	0 – 4	0 = 4 Bits 1 = 5 Bits 2 = 6 Bits 3 = 7 Bits 4 = 8 Bits
Stop Bits	32	0 – 2	0 = 1 Bit 1 = 1,5 Bits 2 = 2 Bits

N.B. Values in red are those currently supported by hardware. Use of other values will render hardware inoperable.

Table 5.3.1: A summary of program parameters along with their respective valid data field values.

Table 5.3.1 summarises program parameters and valid data field values associated with each parameter. Each data field within a program parameters file is checked for valid values when the file is opened and any errors encountered displayed by the panel at the bottom of the main dialog. An error message takes the following format:

File Open Error: Error Type "Data Field"

Table 5.3.2 summarises the three error types that may be encountered and the cause of the error in each case.

Error Type	Cause
Non Numeric Data Value "Data Field"	ASCII character other than 0-9 encountered in the data field
Data Field Value Invalid "Data Field"	Data field value outside allowable range as defined in table 5.3.1
Empty Data Field "Data Field"	Data field is empty

Table 5.3.2: A summary of the three types of error message that may be encountered.

Selection of the 'Communication' category produces a drop down menu, figure 5.3.11 with 'Serial Port Settings' and 'Send Data' options.

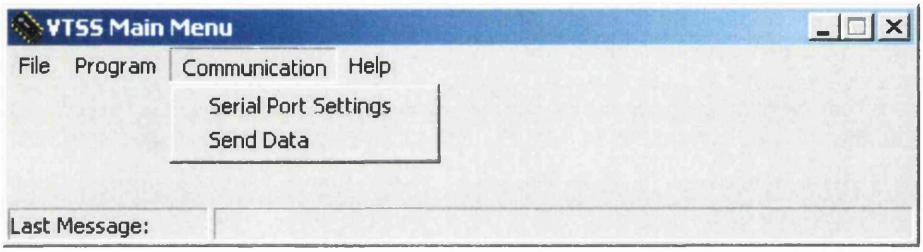


Figure 5.3.11: Communication menu in the main dialog.

Choice of the 'Serial Port Settings' option produces a dialog from which the serial port may be configured, figure 5.3.12. A shareware Delphi serial port tool was employed to provide configuration and data transmission utilities, minimising time investment required to provide a reliable serial communications code.

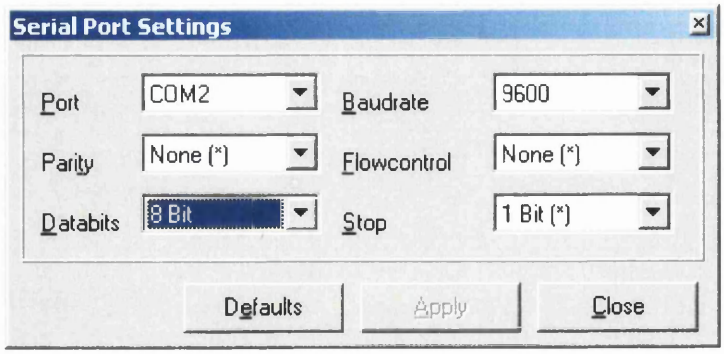


Figure 5.3.12: Serial Port Settings dialog.

The 'Serial Port Settings' dialog allows selection and configuration of serial communications ports COMM1 to COMM8 (if installed). Currently the serial interface implemented by the microcontroller is configured for: 9600 baud rate, no

parity checking, no flow control, 8 data bits and 1 stop bit. To maintain maximum flexibility in regard to future development of microcontroller hardware, the facility to change currently fixed serial port parameters has been incorporated. Currently selection of parameter values other than those listed above will result in loss of communication between the microcontroller and PC. The apply button on the dialog becomes available following a parameter change and must be selected before a new communications configuration is implemented. The parameters selected by toggling the 'Defaults' button are the microcontroller defaults with COMM2 selected.

Choice of the 'Send Data' option produces a confirmation dialog, figure 5.3.13. The 'Send Data' option is only available following selection of either 'Linear' or 'Non-Linear' program options from the 'Program' menu.

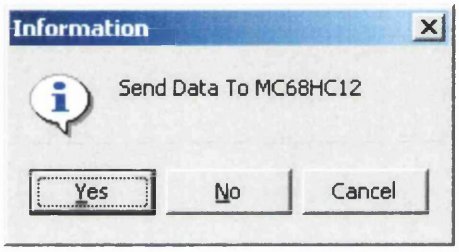


Figure 5.3.13: Send Data conformation.

Selection of the help category produces a drop down menu with 'About' as the only option, figure 5.3.14. Choice of the about option produces an about dialog, figure 5.3.15.

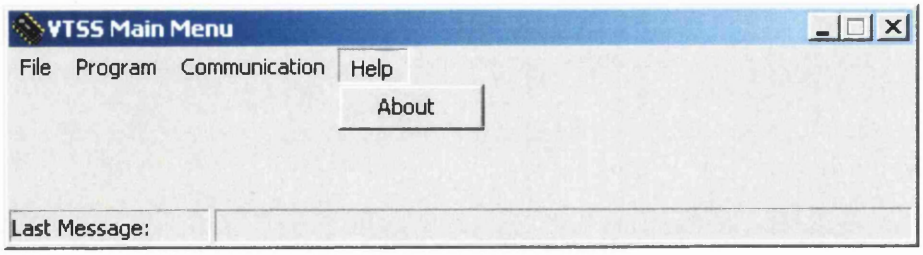


Figure 5.3.14: Help menu in the main dialog.

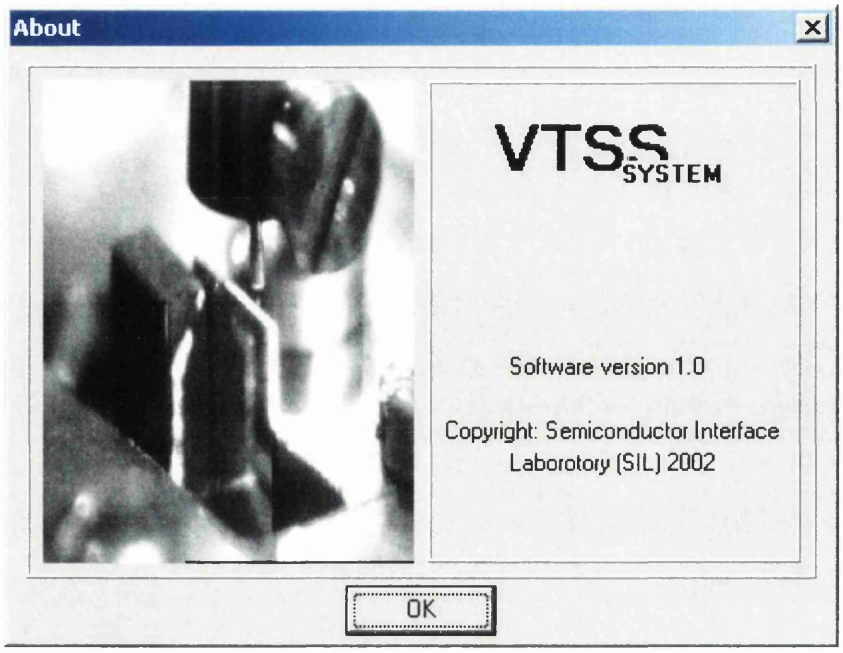


Figure 5.3.15: About dialog.

5.4 Development and description of macros using the SCALA Pro nano-structuring language.

Here a description of the macros composed within the SCALA Pro nano-structuring language to carry out variable and fixed tip sample separation spectroscopy measurements is undertaken. Macros are written in a suitable text editor (Microsoft Wordpad was used to develop macros described here) and saved with the .MAC extension. Spectroscopy macros are loaded within the Spectroscopy Preset window of the SCALA Pro GUI. Figure 5.4.1 shows a screen shot of the Spectroscopy Preset window during selection of a macro routine. Syntactical errors within a macro are revealed when the macro is loaded. For a detailed description of both the nano-structuring language and Spectroscopy Preset refer to [10], [11]. Code listings for the macros described here are located in appendix B.

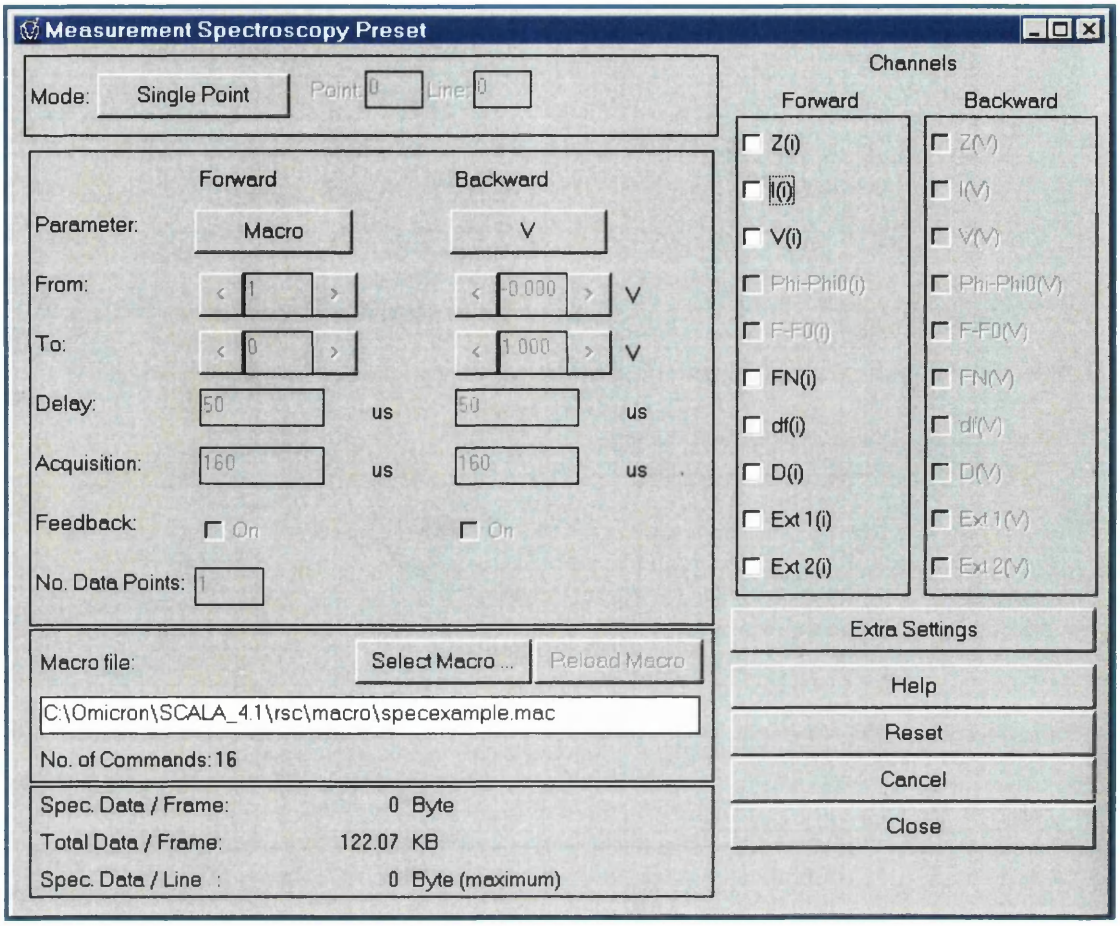


Figure 5.4.1: Screen shot illustrating selection of a macro routine within the spectroscopy preset window.

The macros implement both software and hardware based variable tip-sample separation STS solutions. In the first case the solution is entirely implemented within the nano-structuring language. In the second case software control of external hardware is achieved through four user programmable output signals provided by the SCALA electronics status port. In addition, a macro to perform fixed tip sample separation spectroscopy measurements was developed to provide a means of comparing and evaluating data obtained during measurements. Sections 5.4.1 and 5.4.2 describe macros implementing software and hardware based variable tip-sample separation STS respectively, and section 5.4.3 fixed tip-sample separation STS.

5.4.1 Macro implementing software based variable tip-sample separation STS.

The macro routine implementing software based variable tip-sample separation STS consists of three sections, responsible for measurements as the tip approaches the sample in discrete steps, when it reaches its peak displacement from the regulated tip height, and as it is retracted in discrete steps from sample surface. Figure 5.4.2 illustrates the operations performed by the macro routine to accomplish software based variable tip-sample separation STS.

On entry to the first section of the measurement the tunnel current was measured following a delay. The gap voltage was first decreased by a predetermined offset and the tunnel current measured, then increased by twice the offset and the tunnel current measured again. This effectively allows the conductivity at a specific tip sample separation to be ascertained. The tip sample separation was altered by subtracting an offset from the tip height, which has the effect of moving the tip toward the sample surface. The measurement was repeated within a loop construct for (n-1) / 2 times, where n was equal to the number of data points within the measured spectra.

On exit from the first loop construct the second stage in the measurement proceeds, at which point the tip sample separation has reached the maximum displacement from the regulated tip height. Here the measurements as described above are repeated without altering the tip sample separation.

On entry to the third section of the measurement the actions executed within the first loop are repeated, with the exception that an offset is added to the tip height thus increasing the tip sample separation.

5.4.2 Macro implementing hardware based variable tip-sample separation STS.

The macro routine responsible for implementing hardware based variable tip-sample separation STS consists of a single loop in which status port user signals are set to control the sequence of measurement events. Figure 5.4.3 illustrates the operations performed by the macro routine to accomplish hardware based variable tip-sample separation STS.

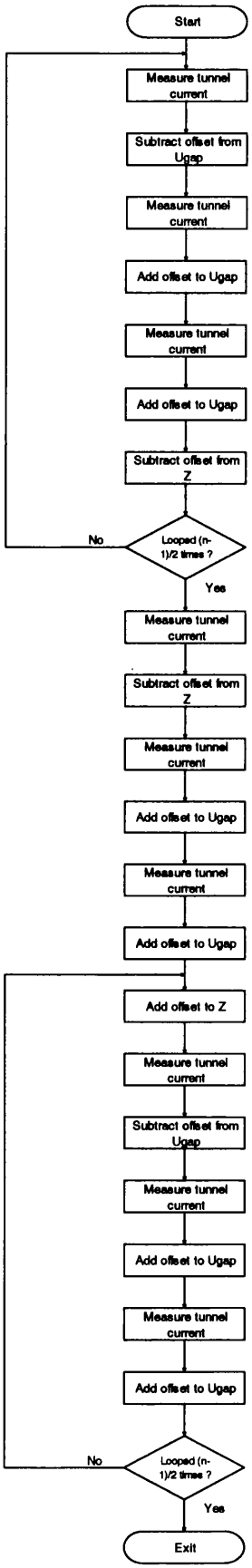


Figure 5.4.2: A Flow chart illustrating the operations performed by the macro routine responsible for software based Variable tip-sample separation STS.

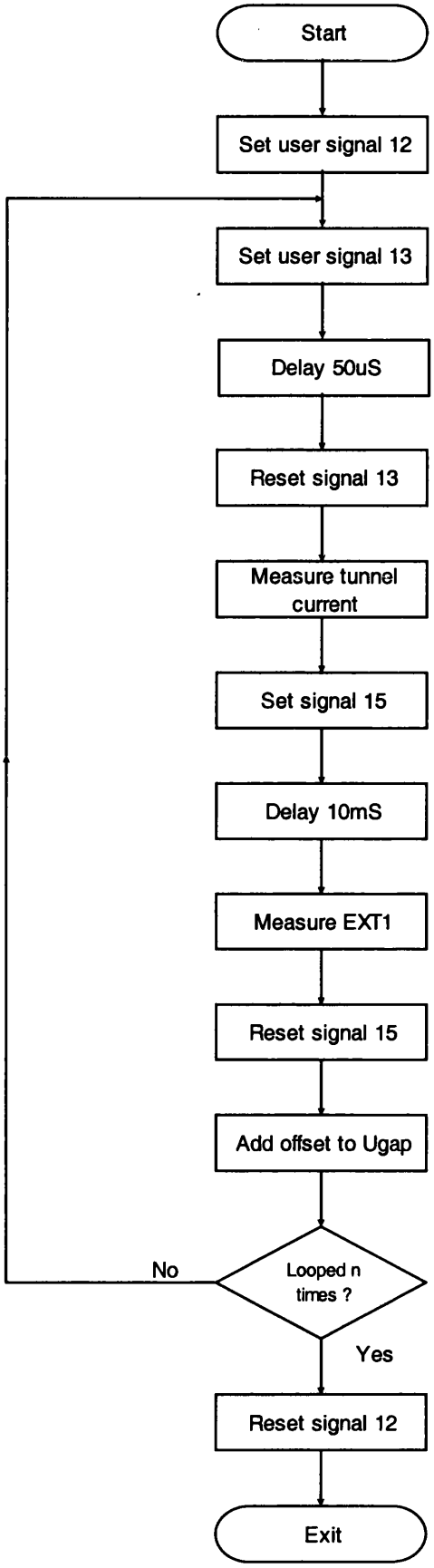


Figure 5.4.3: A flow chart illustrating the operations performed by the macro routine responsible for hardware based variable tip-sample separation STS.

Prior to entry to the loop construct, status port user signal 12 was set to indicate to external hardware that a measurement sequence was to commence. On entry to the loop construct user signal 13 was set for 50uS and then reset, indicating to external hardware that the tip should be moved a single step, after which the tunnel current was measured. Status port user signal 15 was set to route an external modulating signal to the SCALA electronics external gap voltage input, therefore allowing conductivity measurements to be performed by a Lock-in amplifier. The modulating signal is present for a period of time, allowing the output of the Lock-in amplifier to stabilise. The resulting output from the Lock-in amplifier, which was proportional to the conductivity of the sample, was measured using one of the external measurement channels (EXT1) the SCALA electronic possess. User signal 15 was then reset and the gap voltage incremented by a predetermined value. The loop was repeated for n iterations, where n was the number of data points in a measured spectra. After n iterations the loop terminates and user signal 12 was reset, indicating the completion of the measurement sequence.

5.4.3 Macro implementing fixed tip sample measurements.

The macro routine responsible for implementing fixed tip-sample separation STS consists of a single loop construct in which the tunnel current was measured and a predetermined offset added to the sample bias. The loop was repeated for n iterations, where n was the number of data points in a measured spectra. Figure 5.4.4 illustrates the operations performed by the macro routine to accomplish fixed tip-sample separation STS.

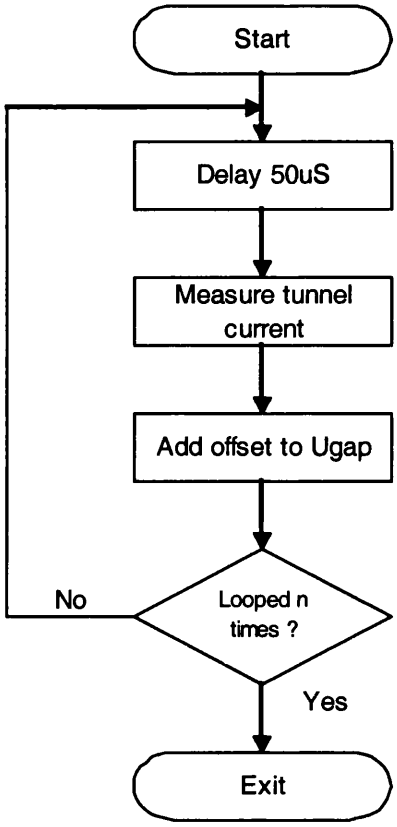


Figure 5.4.4: A flow chart illustrating the operations performed by the macro routine responsible for fixed tip-sample separation STS.

5.5 References.

[1] Motorola, M68HC12B Family Technical Data, Section 18 – Development Support, **438** – **463**, Rev 4.0.

[2] P&E Microsystems web site, PEMICRO.COM.

[3] Motorola, M68HC12B Family Technical Data, Rev 4.0.

[4] Motorola,CPU12 Reference Manual, Rev 3.0.

[5] Analog Devices, AD669 data sheet, Rev.A.

[6] Motorola, M68HC12B Family Technical Data, Section 12 - Standard Timer Module (TIM), **213** – **238**, Rev 4.0.

[7] Motorola, M68HC12B Family Technical Data, Section 14 – Serial Interface, **290** – **302**, Rev 4.0.

[8] R.Lishner, Delphi In A Nutshell, O'Reilly books, 2000.

[9] X.Pacheco and S.Teixeira, Borland Delphi 6 Developers Guide, SAMS publishing, 2002.

[10] Omicron, The SCALA PRO Software Manual, Version 4, **77 – 84**, June 2001.

[11] Omicron, The SCALA PRO Software Manual, Version 4, **225 – 228**, June 2001.

Chapter 6

Evaluation of Instrument Operation

Introduction

Here the instrumentation designed to operate in conjunction with the Omicron SCALA electronics, to implement the variable tip-sample separation technique was tested as an entire system. In section 6.1 the operation of the instrumentation with the Linear and Non-Linear programs is discussed. In section 6.2 the instrumentation used to perform conductivity measurements is assessed and problems encountered discussed.

6.1 Evaluation of instrumentation operation with the Linear and Non-Linear programs.

In order to access the functionality of the instrumentation responsible for generating the tip-sample separation contour during variable tip-sample separation STS, signals at specific locations were monitored and compared with theoretical expectations. Figure 6.1.1 shows a schematic illustrating the locations where signals were monitored during measurements.

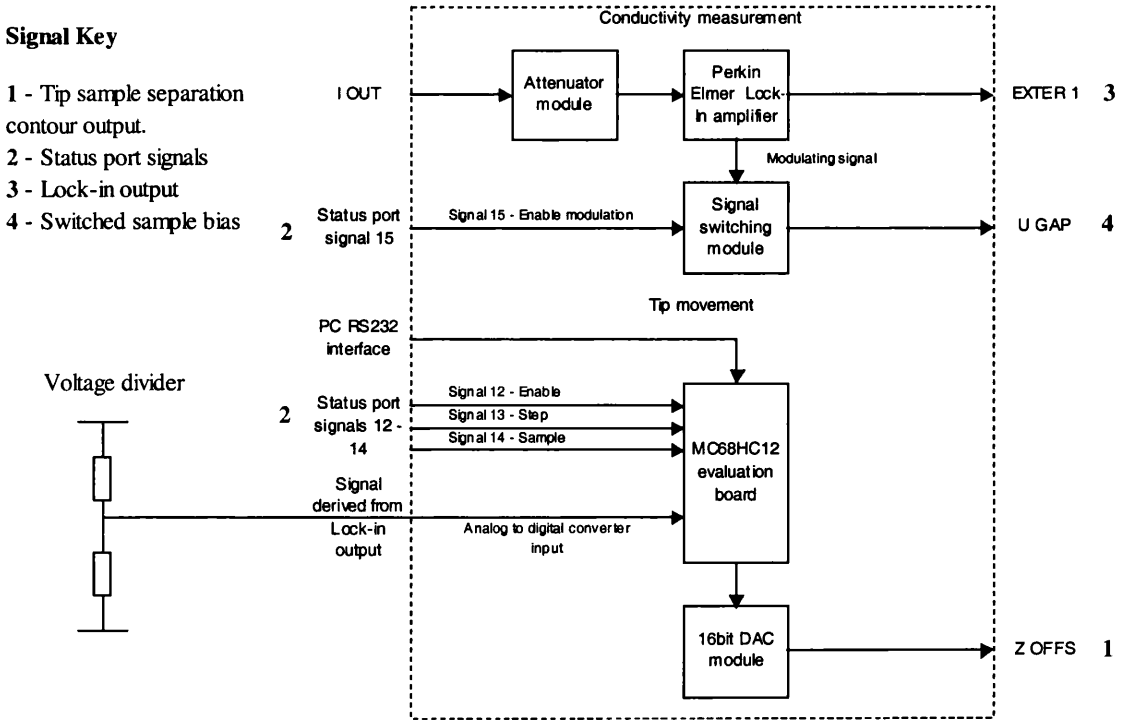


Figure 6.1.1: A Schematic of the instrumentation illustrating locations where signals were monitored during testing.

6.1.1 Operation of the system with the Linear program.

During testing of the instrumentation it was necessary to adjust the amplitude of the tip-sample separation contour, in order to calibrate the system. This was accomplished by performing spectroscopic measurements while adjusting the output amplitude of the DAC module, and monitoring the amplitude of the tip-sample separation contour. The amplitude of the tip-sample separation contour was conveniently assessed by examining the trace of tip displacement from the regulated height provided by the SCALA Pro software, as illustrated by figure 6.1.2.

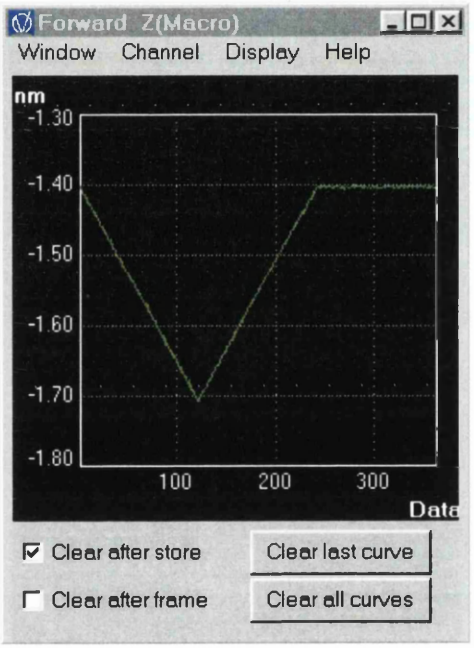


Figure 6.1.2: A screen shot illustrating the display of tip displacement from regulated height as provided by the SCALA Pro software.

Figure 6.1.3 (a) shows an oscilloscope trace of the tip-sample separation contour and (b) the sample bias for a single spectroscopy measurement. These traces were obtained with typical spectroscopy parameters of: -2 to +2V sample bias, 41 data points and tip-sample contour scaling factor of 0.5nm. For each of the 41 steps in sample bias a corresponding step in tip-sample separation is observed. The tip-sample separation achieves a peak value of -98.1mV as the sample bias reaches 0V, corresponding to a tip displacement of 0.5nm from the regulated height. This value is in good agreement with expectations, since the SCALA manual [1] states that 20mV

is required to produce a displacement of 0.1nm. The difference of 1.9mV is attributed to tolerances of components within the SCALA electronics.

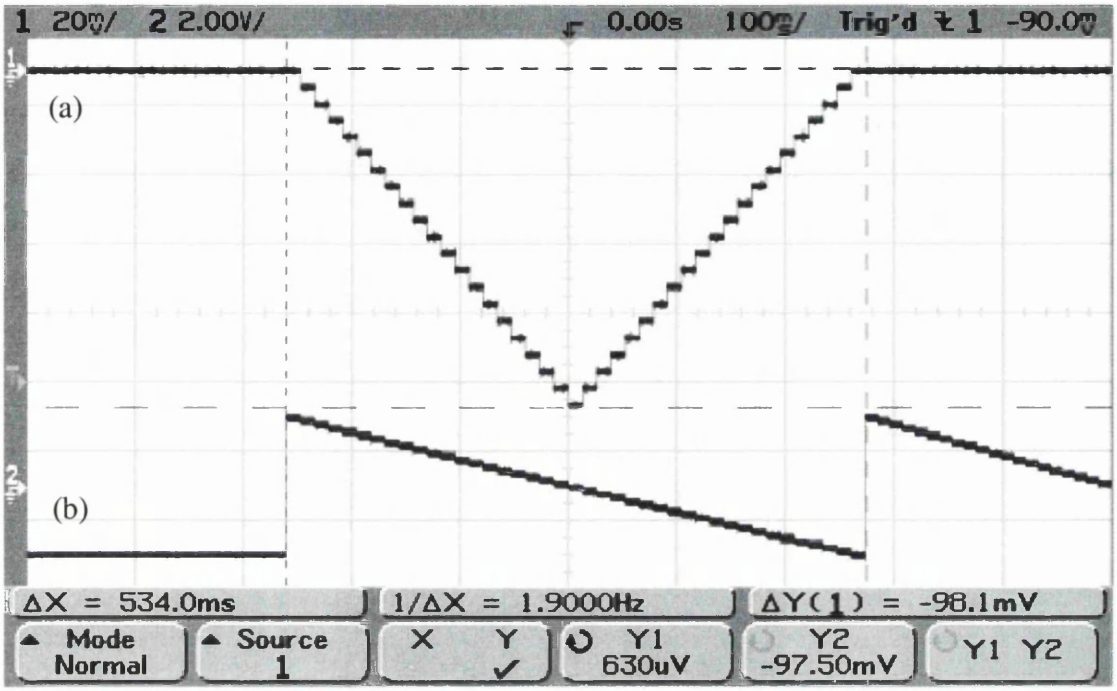


Figure 6.1.3: Oscilloscope traces of (a) the tip-sample separation contour and (b) the sample bias during a single spectroscopy measurement.

To orchestrate the order in which events occur during spectroscopy, signals 12 and 13 provided by the SCALA status port were exploited. These signals synchronise events within spectroscopy macros with the external instrumentation; signal 12 functions as an enable signal and signal 13 as a means of indicating a change in tip-sample separation is required. When signal 12 is asserted low, transition of signal 13 from high to low will result in a change of tip-sample separation. This situation is observed in figures 6.1.4 and 6.1.5, which illustrate the relationship between signals 12 and 13, the sample bias and the tip-sample separation contour. It is observed that when signal 12 is asserted low, a pulse of ~50uS in signal 13 following a step in sample bias results in a corresponding step in tip-sample separation. When signal 12 is inactive the tip-sample separation is zero irrespective of the state of the sample bias and signal 13.

To ensure correct operation, the system was exercised with a range of parameters that reflected those likely to be encountered during normal operation. Throughout these tests the system performed as expected.

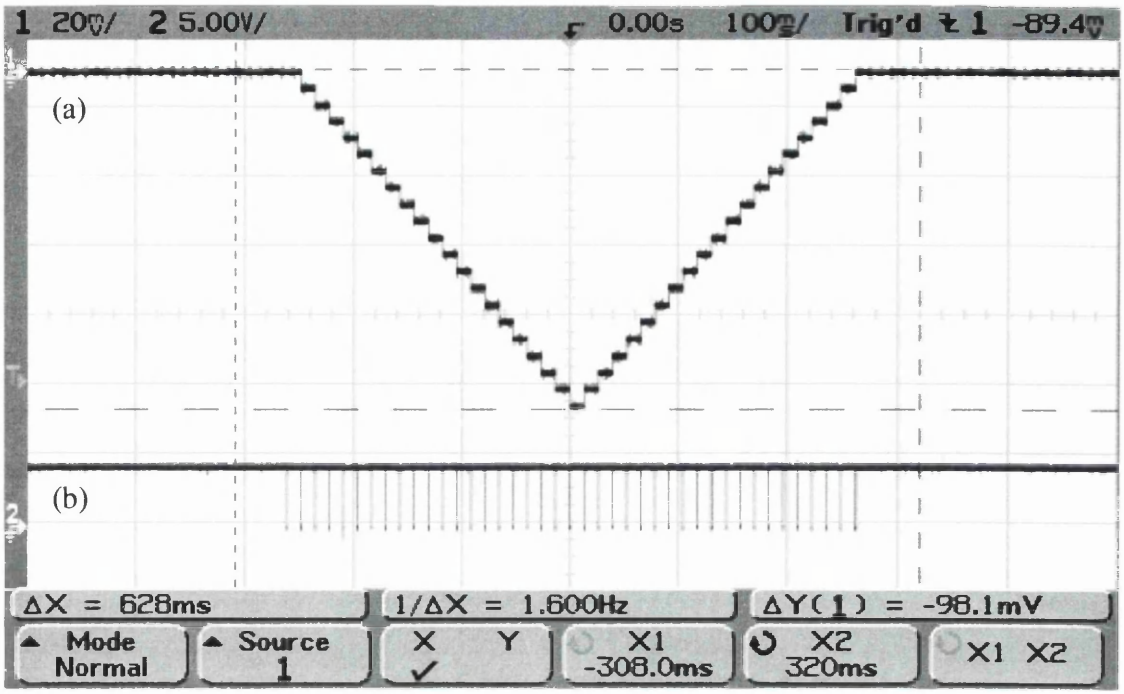


Figure 6.1.4: Oscilloscope traces of (a) the tip-sample separation contour and (b) status port signal 12 during a single spectroscopy measurement.

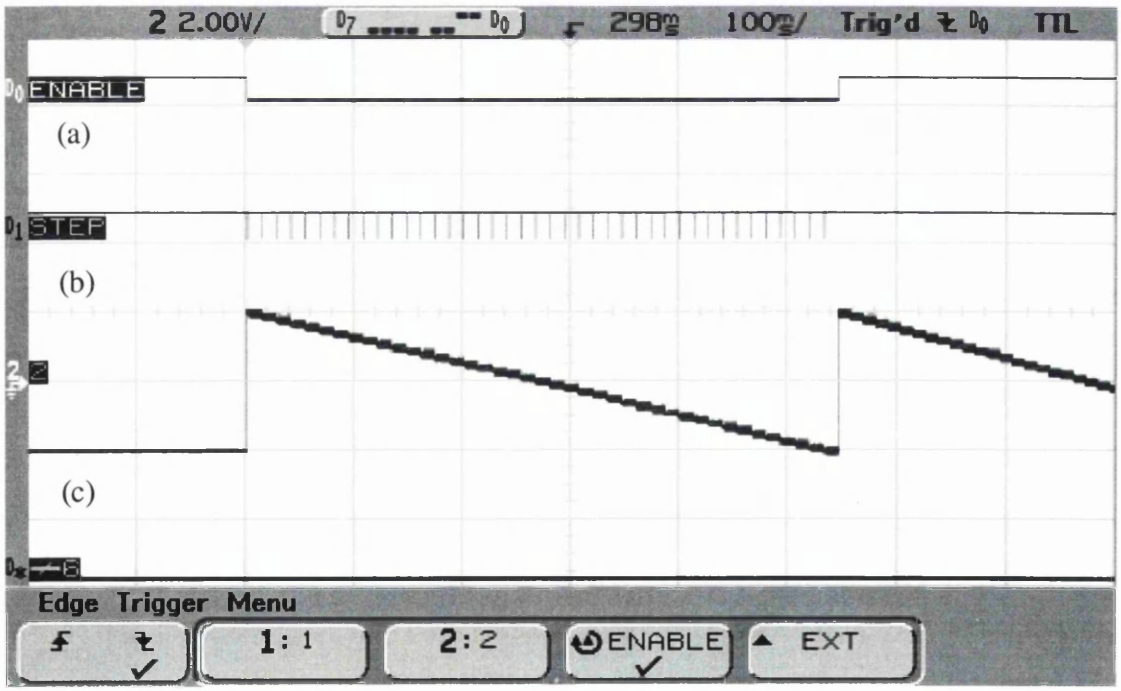


Figure 6.1.5: Oscilloscope traces of (a) status port signal 12 also labelled 'ENABLE' and (b) status port signal 13 also labelled 'STEP' and (c) the sample bias during a single spectroscopy measurement.

6.1.2 Operation of the system with the Non-Linear program.

To evaluate the operation of the system with the Non-Linear program, a range of program parameters were selected and spectroscopy performed with a steady voltage supplied to Port AD0 of the microcontroller. The voltage supplied to Port AD0 was provided by a voltage divider circuit as shown in figure 6.1.1, which supplies a steady and variable voltage between 0 and 5Vdc. This voltage emulates the output of circuitry responsible for measuring sample conductivity, and determines the amplitude of the next step in the tip-sample separation contour. To sample the voltage, status port signal 14 is asserted low for 50uS during the period of time conductivity measurements are performed for each data point. When the voltage is 5Vdc the amplitude of the step is equal to the maximum step size defined in the Non-Linear program dialogue, which can take on values in the range 0.05nm to 0.2nm in 0.05nm increments.

The maximum tip displacement from the regulated tip height during variable tip-sample separation STS is constrained to be in the range 0.05nm to 0.95nm. The choice of the maximum tip displacement is defined by the Z-limit field within the Non-Linear program dialogue.

To access the operation of the Non-Linear program, the system was exercised with a range of the parameters spanning the complete operating range, whilst the voltage applied to Port AD0 was adjusted in suitable increments. Throughout these tests the instrumentation behaved as expected.

As an example, figure 6.1.6 shows an oscilloscope trace of the tip-sample separation contour generated by the Non-Linear program. The trace was obtained with typical spectroscopy parameters of: -2 to +2V sample bias, 41 data points, a maximum allowable tip displacement from the regulated tip height of 0.5nm and a maximum step size of 0.05nm.

In this case, to access the effect of varying the voltage supplied to Port AD0, the voltage was adjusted in 0.5V increments from 0 – 5Vdc and spectroscopy performed after each adjustment. As the voltage was increased a corresponding increase in the

step size was observed, reaching ~ 0.05nm when the applied voltage reached 5Vdc. The maximum displacement of the tip from the regulated tip height was constrained to 0.5nm as expected.

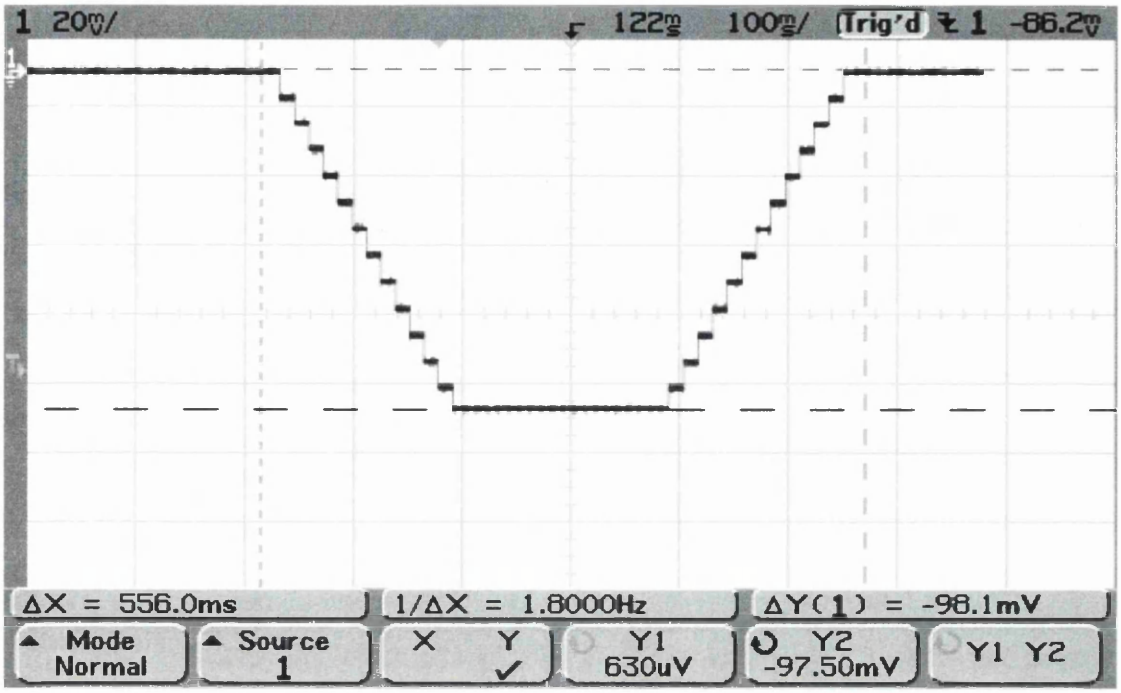


Figure 6.1.6: An oscilloscope trace of the tip-sample separation contour produced by the Non-Linear program during a single spectroscopy measurement.

6.2 Conductivity measurements.

The variable tip-sample separation technique requires that measurements of conductivity be performed for every discrete change in tip-sample separation [2]. To accommodate this, conductivity measurements were performed for every data point in a spectra. To perform conductivity measurements one of two methods is adopted: the first and simplest method was implemented without the necessity of external instrumentation, whereas the second method utilises a Lock-in amplifier. Both methods were utilised by spectroscopy macros composed to implement the variable tip-sample separation technique. A detailed description of both the aforementioned methods is given in chapters 4 and 5 and not repeated here. Discussion is limited here to testing of and parameter selection for the external instrumentation involved in the implementation of the second method.

To carry out conductivity measurements the Lock-in amplifier provides a stable and pure sinusoidal reference signal that was used to modulate the sample bias. This modulation was applied to the sample bias for a period of time for each data point, following measurement of the tunnel current. Modulating the sample bias produces a corresponding sinusoidal component in the tunnelling current at the reference signal frequency. The Lock-in amplifier extracts the tunnel current component at the reference signal frequency and effectively differentiates it with respect to the reference signal, producing a voltage that is proportional to the conductivity, which was recorded.

The sequence and chronology of the combined tunnel current and conductivity measurement for a single data point is illustrated in figure 6.2.1. Figure 6.2.1 (a) shows the enable signal (status port signal 12) which was asserted low for the duration of the measurement, indicating to the microcontroller an STS measurement was in progress. Figure 6.2.1 (b) shows the step signal (status port signal 13) which was asserted low for 50uS following a change in sample bias. This indicates to the microcontroller that a change in tip-sample separation was required. The tunnel current was measured following a short delay after the sample bias changes. After another short delay the modulating signal was superimposed onto the sample bias, as illustrated by figure 6.2.1 (c). Superimposition of the modulating signal on the sample bias was accomplished by switching the reference signal provided by the Lock-in amplifier using the Signal Routing module, and applying it to the external sample bias input provided by the SCALA electronics.

Figure 6.2.2 (a) shows an oscilloscope trace of the tip-sample separation during acquisition of variable and fixed tip-sample separation spectra, and figure 6.2.2 (b) the conductivity spectra generated by the Lock-in amplifier for variable and fixed tip-sample separation respectively. The conductivity spectra were obtained with the modulation present for 10ms per data point, leading to a total measurement time of 10s with 121 data points per spectra.

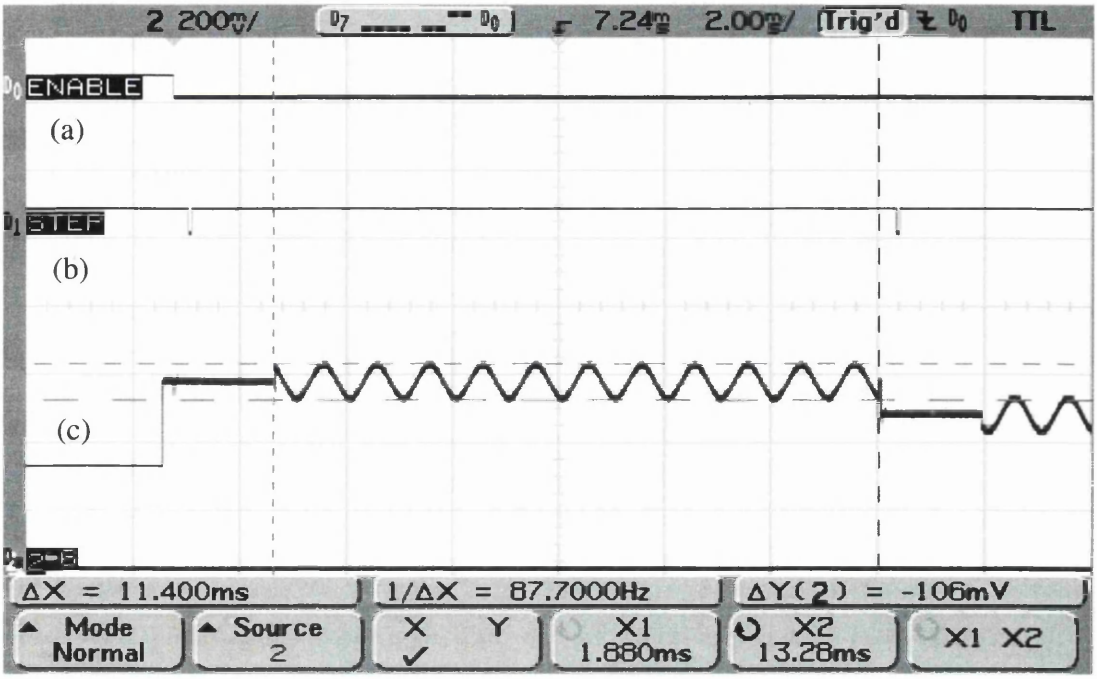


Figure 6.2.1: Oscilloscope traces showing (a) the enable signal (status port signal 12) (b) the step signal (status port signal 13) and (c) the sample bias for a single data point.

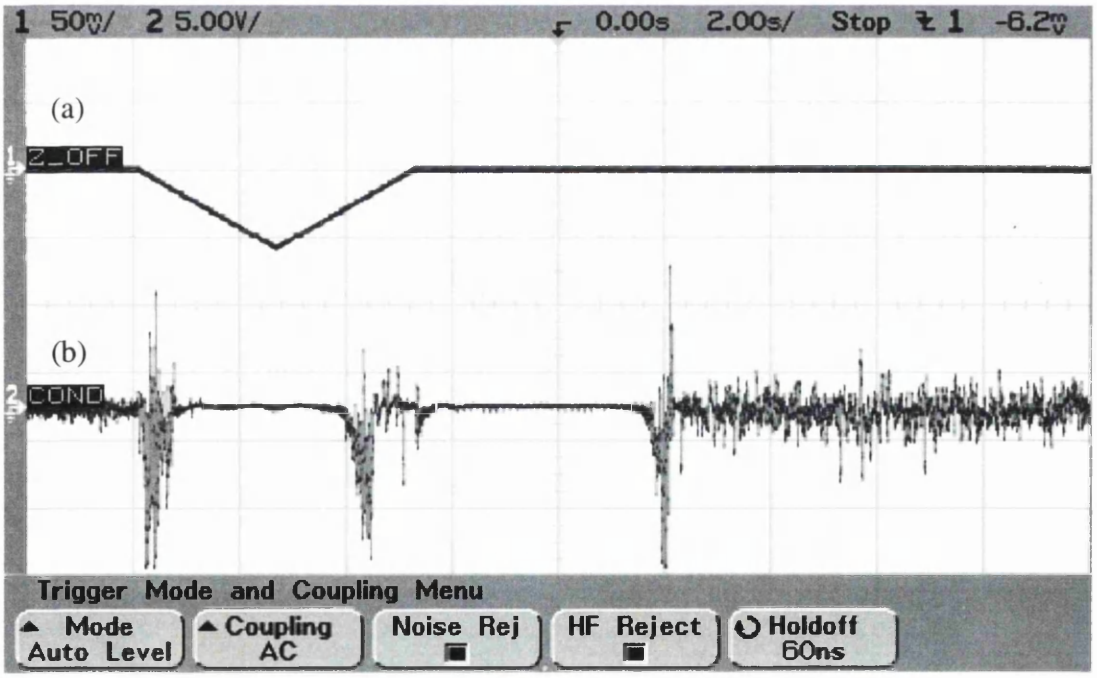


Figure 6.2.2: Oscilloscope traces of (a) the tip-sample separation during acquisition of variable and fixed tip-sample separation spectra and (b) the conductivity spectra for variable and fixed tip-sample separation.

The selection of Lock-in parameters that defined attributes of the reference signal and governed recovery of the tunnel current component, had a pronounced effect on the quality of the conductivity spectra acquired. These parameters varied significantly between samples and required tailoring to an individual sample to obtain the best results. The discussion is limited here to generalities regarding selection of these parameters and problems encountered whilst measurements were made.

The initial parameters selected for the reference signal were a frequency of 1KHz and an amplitude of 100mV Pk-Pk. The choice of the initial frequency was decided upon from typical values used by other authors [2], [3]. When measurements were performed it was necessary to adjust the amplitude of the reference signal depending on the conductivity of the sample. For samples with higher conductivities the reference signal amplitude was reduced to 10mV, which produced a reasonably sized tunnel current component.

The initial Lock-in parameters selected to extract the component of the tunnel current caused by the modulation of the sample bias were a time constant of 10ms, a sensitivity of 200mV and phase of 0^{o}. Here the time constant refers to the time the Lock-in requires to arrive at a stable output, and the sensitivity the input amplitude required to produce a full scale output, and the phase the difference between reference signal phase and the input signal phase. More information concerning the function of the parameters discussed above is available from the Lock-in amplifier reference manual [4].

When conductivity measurements were performed with the initial parameters the conductivity spectra incorporated artefacts not present in the tunnel current spectra, as shown in figure 6.2.3 (a) and (b) respectively. These artefacts were introduced by the Lock-in during the measurement.

As the time the reference signal was present for during each data point was increased there was a corresponding reduction in the amplitude of the artefacts within the spectra, however the measurement time for the complete spectra was increased accordingly. This increase in measurement time increased the sensitivity of the measurement to thermal drift, therefore introducing uncertainties as to the physical

location of the measurement on the sample surface. In order to minimise both the measurement time and components introduced into the spectra by the Lock-in amplifier, the number of data points within the spectra was reduced. This however leads to a reduction in the resolution of the spectra without significant reduction in measurement time. The precise origin of the artefacts introduced by the Lock-in is uncertain, however in order to achieve sufficient resolution within the spectra long measurement times were necessary, which means measurements on atomic scale features are difficult using this method.

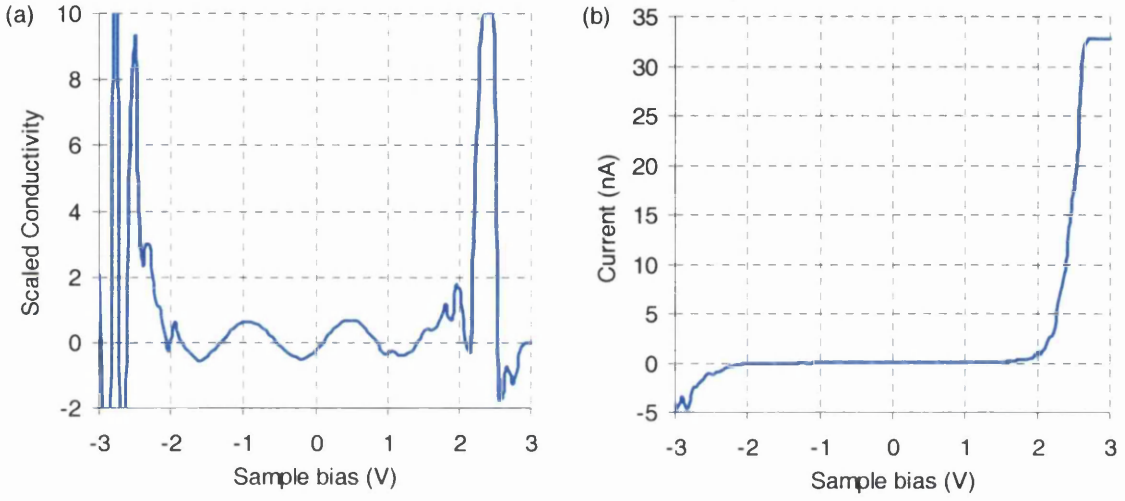


Figure 6.2.3: (a) Conductivity and (b) Tunnel current spectra for a variable tip-sample separation STS measurement. The conductivity spectra incorporates artefacts introduced by the Lock-in amplifier.

It was observed that as the frequency of the reference signal was increased there was a consistent increase in the amplitude of the tunnel current component at 90° to the reference signal phase. This is illustrated by table 6.2.1 which shows a record of the Lock-in output amplitude for phases of 0°and 90° over a range of frequencies.

The tunnel current component displaced from the reference signal phase by 90° is caused by the capacitance of the tip-sample system [3]. As the reference signal frequency was increased the capacitive component manifested itself as an increasing offset in the conductivity spectra. Although this offsetting effect is reduced at the lowest reference signal frequency, it is probable that the tip-sample capacitance

perturbs the conductivity measurement; therefore the conductivity spectra probably contain artefacts introduced by the tip-sample capacitance.

Frequency (KHz)	Phase (°)	Output (mV Pk-Pk)
1	0	2.83
1	90	10.75
2	0	3.39
2	90	28.29
3	0	11.88
3	90	41.58
4	0	23.48
4	90	52.33
5	0	36.78
5	90	60.86

Table 6.2.1: A table showing the Lock-in output at 0° and 90° for a range of reference signal frequencies.

The factors outlined above lead to adoption of the first conductivity measurement method in preference to second method using the Lock-in amplifier.

6.3 References.

[1] Omicron NanoTechnology, 'The SCALA PRO Software Manual', Version 4.1 (2001).

[2] J.A. Stroscio, R.M. Feenstra, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 4, **135-138**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[3] R.J. Hamers, D.F. Padowitz, 'Scanning Probe Microscopy and Spectroscopy', Chapter 4, **73**, Edit D.A. Bonnel, Second edition, Wiley-VCH (2001).

[4] Perkin Elmer, Model 7265 DSP Lock-in Amplifier, instruction manual, 2000.

Chapter 7

STM and STS studies of Si(111) 7x7 and GaAs(110)

Introduction

Results are presented here for STM and fixed and variable tip-sample separation STS measurements on Si(111) 7x7 and p-type GaAs(110). Results on Si(111) 7x7 are presented in section 7.1 and on p-type GaAs(110) in section 7.2. In sections 7.1 and 7.2 sample preparation techniques are first addressed followed by discussion of STM results and finally discussion of fixed and variable tip-sample separation STS results. The objective of the tests was to access the instrumentation operation on well understood surfaces, and compare the results with previous work.

7.1 Application of STM and fixed and variable tip-sample separation STS to p-type Si(111) 7x7.

7.1.1 Preparation of Si(111) 7x7.

To prepare a Si(111) 7x7 sample for STM a rigorous procedure is adopted that ensures the sample is free from contamination prior to scanning. A sample 9mm in length and 2mm wide was cut from a p-type Si(111) wafer, doped with B at a concentration of ~3 x 10^{16} cm^3, by scribing the surface and then fracturing along the scribe mark. Debris remaining from the cutting process was removed by passing a flow of dry nitrogen across the sample followed by an ultrasonic bath whilst immersed in acetone. Throughout processing the Si sample was handled using plastic tools to avoid contamination with Ni from stainless steel tools.

The Si sample was clamped into a sample plate, as illustrated by figure 7.1.1, the clamps form electrical connections through which a current can be passed, facilitating self heating. This type of heating is termed 'direct current' and provides a method whereby the sample can be cleaned. An optical pyrometer fixed to the preparation chamber was focussed on the Si sample and used to provide an accurate temperature measurement.

The Si was outgassed by direct current heating at 580°C for 12hours. Initially the power applied to the sample was gradually increased, ensuring that the pressure in the prep chamber did not exceed 10^{-10} mbar. Following outgassing, the native oxide was

removed by momentarily direct current heating the sample to 1250°C, whilst ensuring the pressure within the preparation chamber remained below 10^{-9}mbar. The sample was left to cool to room temperature and transferred to the STM chamber.

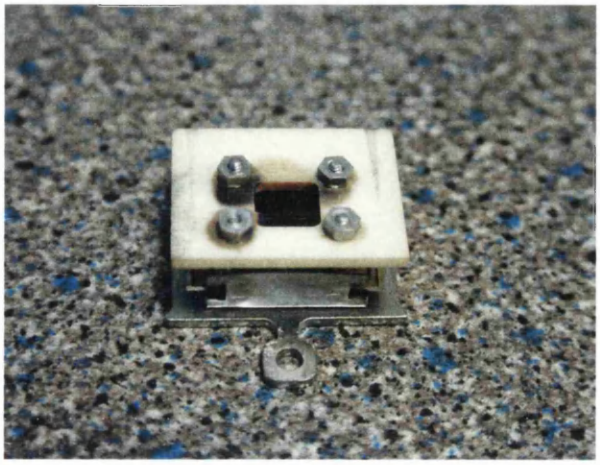


Figure 7.1.1: A photograph showing the sample plate.

7.1.2 STM of Si(111) 7x7

The 7x7 reconstruction of Si(111) has been studied extensively since the 1950's using various surface science techniques. The first images of Si(111) 7x7 obtained by STM were reported by Binning and Rohrer in 1983 [1], aiding the development of the widely accepted dimer adatom stacking fault (DAS) model proposed by Takayanagi *et al* [2]. The 7x7 reconstruction of the Si(111) surface consists a unit cell comprising 49 surface atoms. Figure 7.1.2 shows an STM image of Si(111) 7x7 obtained with positive sample bias. The boundaries of the unit cell are marked, connecting 4 deep depressions at the corners and enclosing 12 protrusions centred on the adatoms of the DAS model [2]. Labels 'F' and 'U' denote the faulted and un-faulted halves of the unit cell, each consisting of a triangular subunit, appearing symmetrical about the short diagonal of the unit cell.

When STM images are acquired with positive sample bias, tunnelling occurs from the tip into empty surface states, and the adatoms within the unit cell appear uniform in the faulted and un-faulted subunits. Figure 7.1.3 shows an STM image of Si(111) 7x7 and a line profile across the long diagonal of a unit cell obtained with positive sample bias. The peaks labelled 1 to 4 centred on adatoms in the faulted and un-faulted subunits differ only slightly in magnitude.

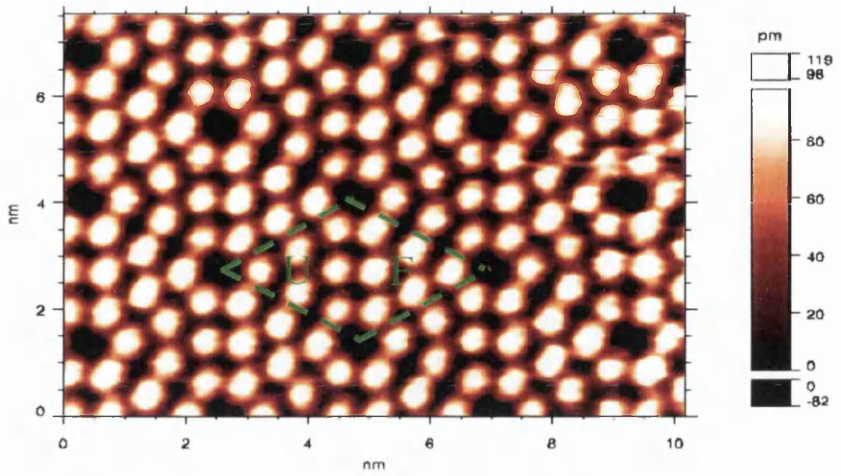


Figure 7.1.2: An STM image of the surface of p-type Si(111) 7x7 with the unit cell indicated. Labels 'F' and 'U' refer to the faulted and un-faulted halves of the unit cell respectively. Image obtained with 1.8V sample bias and a constant tunnel current of 0.7nA

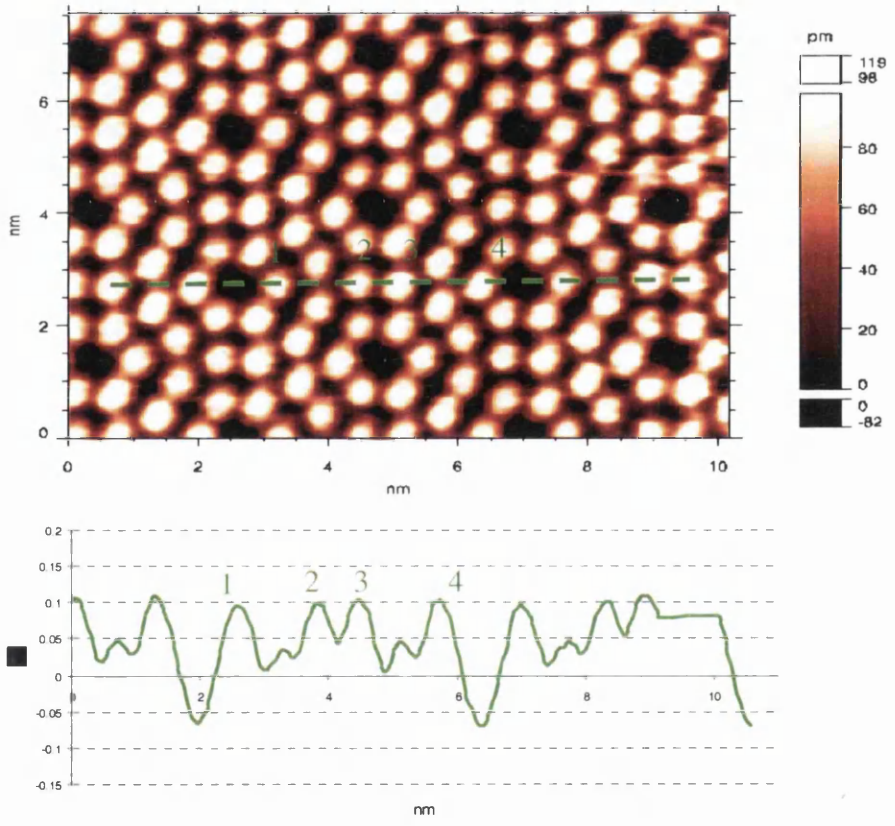


Figure 7.1.3: An STM image of the surface of p-type Si(111) 7x7 and a line profile across the long diagonal of the unit cell obtained with the sample biased at 1.8V and a constant tunnel current of 0.7nA. Points labelled 1 to 4 correspond to peaks in the line profile.

When STM images are acquired with negative sample bias the tunnelling occurs from filled surface states to the tip, and the adatoms within faulted subunit appear higher than those in the un-faulted subunit. Additionally, within the faulted subunit the 3 adatoms nearest the corner holes appear higher than those within the rest of the subunit. Figure 7.1.4 shows an STM image of Si(111) 7x7 and a line profile across the long diagonal of the unit cell obtained with negative bias. The peaks labelled 1 to 4 in figure 7.1.4 centred on the adatoms within faulted and un-faulted subunits differ in magnitude significantly.

The height difference between STM images obtained for positive and negative sample biases shows that an STM image contains both electronic and topological information. At positive sample bias tunnelling occurs from filled states in the tip to empty surface states. Since only a marginal height difference between adatoms in faulted and un-faulted subunits is observed in figure 7.1.3, the density of empty states associated with adatoms in both subunits is similar. At negative sample bias tunnelling occurs between filled surface states in the sample and the tip. The 3 adatoms at the corners of the faulted subunit appear brighter in figure 7.1.4 than the remaining adatoms, indicating a greater density of filled states than their neighbours. The faulted subunit is brighter than the un-faulted subunit, indicating a greater density of filled states associated with the faulted subunit.

7.1.3 Fixed and variable tip-sample separation STS measurements on Si(111) 7x7

Before discussing the spectra obtained using fixed and variable tip-sample separation STS, a brief summary of the work carried out by Hamers *et al* [3, 4, 5] is presented to develop an understanding of the spectral features observed from measurements performed on Si(111) 7x7. Hamers identified the physical origin of surface states arising within spectra obtained from atomically resolved tunnelling spectroscopy measurements carried out on Si(111) 7x7. Figure 7.1.5 (a) shows spatially localised tunnelling spectra and (b), normalised tunnelling spectra averaged over an entire unit cell. States near -0.35V and 0.5V are ascribed to the 12 adatoms within the unit cell, whilst the state at -0.8V arises from the 6 rest atoms and the state at -1.7V is most pronounced near corner holes.

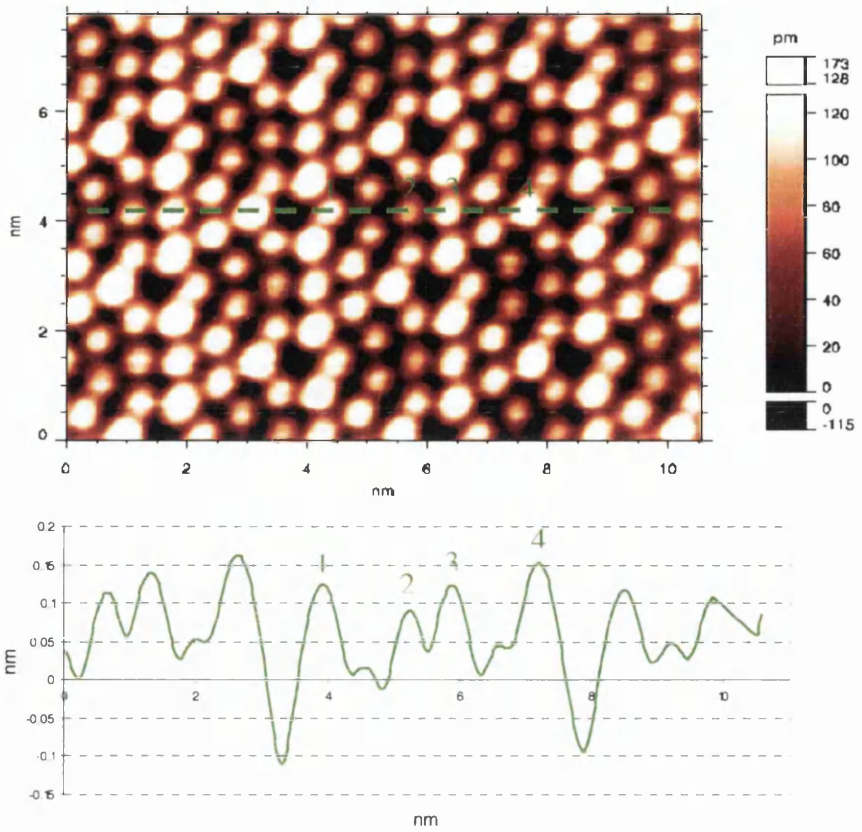


Figure 7.1.4: An STM image of the surface of p-type Si(111) 7x7 and a line profile across the long diagonal of the unit cell obtained with the sample biased at -1.8V and a constant tunnel current of 0.7nA. Points labelled 1 to 4 correspond to peaks in the line profile.

In the study presented here, fixed and variable tip-sample separation spectroscopic measurements on the Si(111) 7x7 samples were performed using a macro composed within the macro language inherent within the SCALA Pro software (See chapters 3 and 5). The macro was structured so that variable tip-sample separation STS measurements are performed first followed by fixed tip-sample separation STS measurements. The offset applied to the regulated tip-height during variable tip-sample separation STS measurements was generated using external hardware, controlled via the macro.

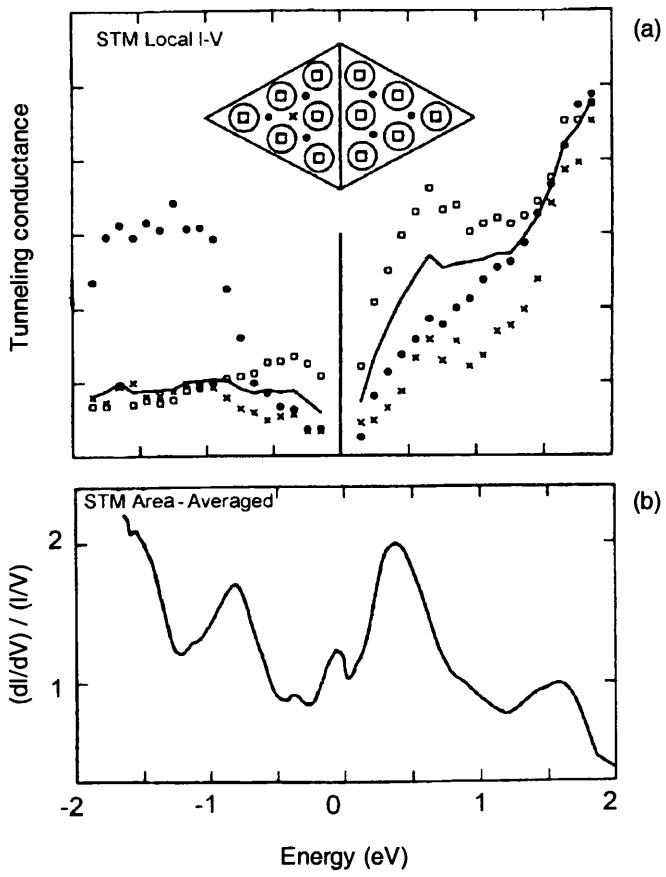


Figure 7.1.5: (a) Spatially localised tunnelling conductivity spectra obtained on the Si(111) 7x7 surface and (b) normalised tunnelling spectra averaged over an entire unit cell. Taken from [5]

The number of data points within a single spectra was selected to give sufficient resolution in that spectra, whilst ensuring the measurement time was not excessive. Generally the aforementioned constraints were achieved with 50mV increments in bias voltage; therefore 121 data points were required for a sample bias range of -3 to +3V.

An acquisition time of 1280us, which is equivalent to 128 samples per data point produced smooth spectra. It was necessary to ensure measurement times were not excessive in order to avoid introducing inaccuracies in the measurement due to thermal drift.

Conductivity measurements, necessary at each discrete step in bias voltage for the variable tip-sample separation method, were performed by modulating the bias

voltage and recording the resultant modulation in tunnel current. Physically this was accomplished by adding an offset in the range 5-50mV to the bias voltage and measuring the resulting tunnel current, then subtracting twice the offset from the bias voltage and repeating the measurement. The value of offset selected depended on the conductivity of the sample and was tailored to result in a measurable change in tunnel current.

During topographic imaging the macro was executed at sites of interest within the Si(111) unit cell. For each execution of the macro a corresponding numbered curve was produced and stored. Stored curves were exported in ASCII text format from the SCALA Pro software and processed using Microsoft Excel. Information regarding the specific structure of spectroscopy files can be located in the SCALA reference manual [6], since a detailed description of the file structure is not undertaken here. Each curve within the exported file contains 484 data points; the first 363 data points contain the variable tip-sample separation STS data and the remaining 121 the fixed tip-sample separation STS data. The requirement for 363 data points within each variable tip-sample separation spectra arises from the necessity to perform measurements of tunnel current and conductivity for every step in sample bias.

Figure 7.1.6 (a) shows raw current data for fixed (green curve) and variable (blue curve) tip-sample separation measurements after averaging many equivalent adatom sites within the Si(111) 7x7 unit cell. A visual comparison between the curves reveals that the magnitude of the raw current in the variable curve is significantly larger than the fixed curve at comparative sample biases, over the complete sample bias range. The linear tip-sample separation contour, figure 7.1.6 (c) has a gradient of 1Å/V and attains a peak displacement from the regulated height of 3Å. This choice of contour gradient gave good results. An increase in the contour gradient resulted in a corresponding increase in tunnel current magnitude for a given sample bias. Figure 7.1.6 (b) and (d) show the conductivity and normalised conductivity curves for the fixed tip-sample separation respectively. The conductivity curve was calculated directly form the raw current data by numerical differentiation and the normalised conductivity calculated by dividing the conductivity dI/dV by the total conductance I/V.

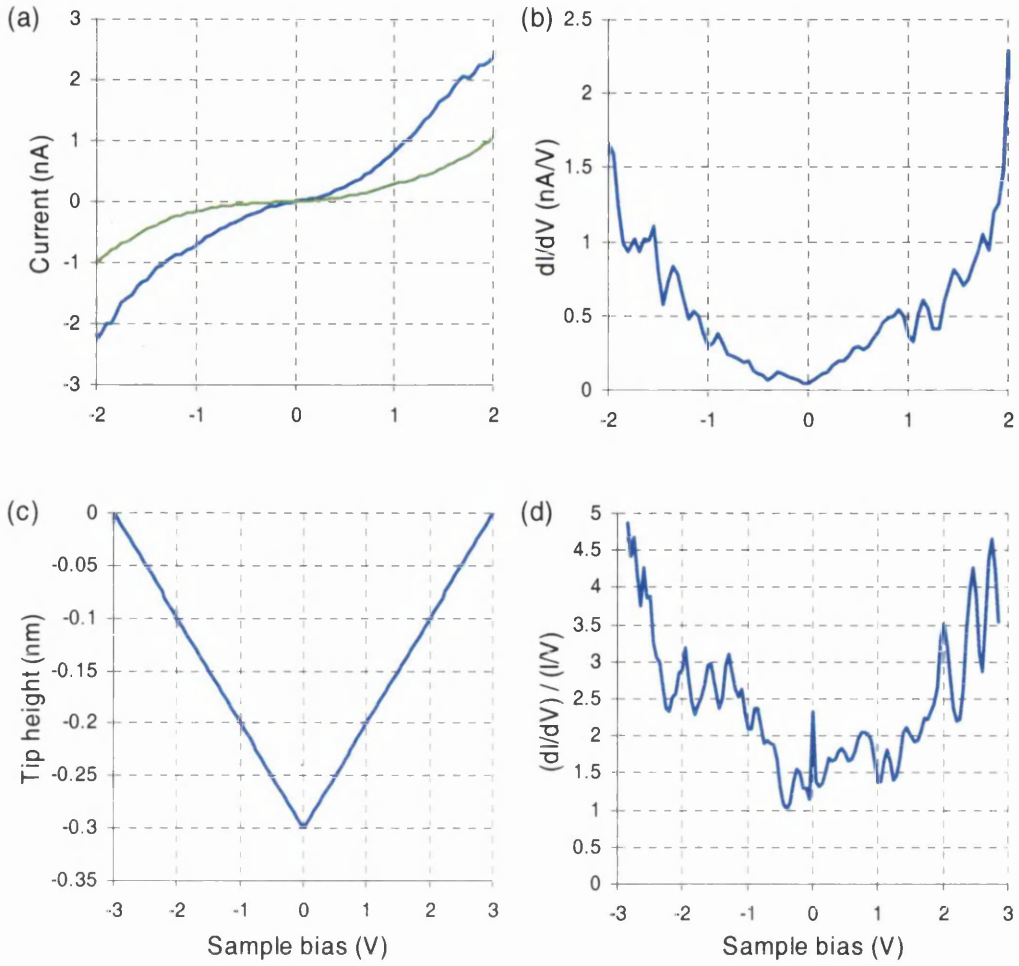


Figure 7.1.6: (a) raw current curves obtained for fixed (green) and variable (blue) tip-sample separation STS measurements. (b) Conductivity calculated from the raw current data for fixed tip-sample separation measurements. (c) Tip-sample separation in reference to the regulated tip height for variable tip-sample separation measurements. (d) Normalised conductivity for fixed tip-sample separation measurements.

The raw current and conductivity data obtained for variable tip-sample separation STS measurements requires processing to remove the effects of varying the tip-sample separation. Transformation of raw data to fixed tip-sample separation equivalency is accomplished by the method described in chapter 2. In the remainder of this section, the application of this method is discussed.

Numerical differentiation of the raw tunnel current to obtain conductivity information, as carried out with the fixed tip-sample separation is not possible, since it yields a total conductivity that includes contributions from varying the tip-sample separation. To obtain conductivity information the sample bias was modulated at each discrete

tip-sample separation and the resultant modulation in tunnel current recorded as described earlier.

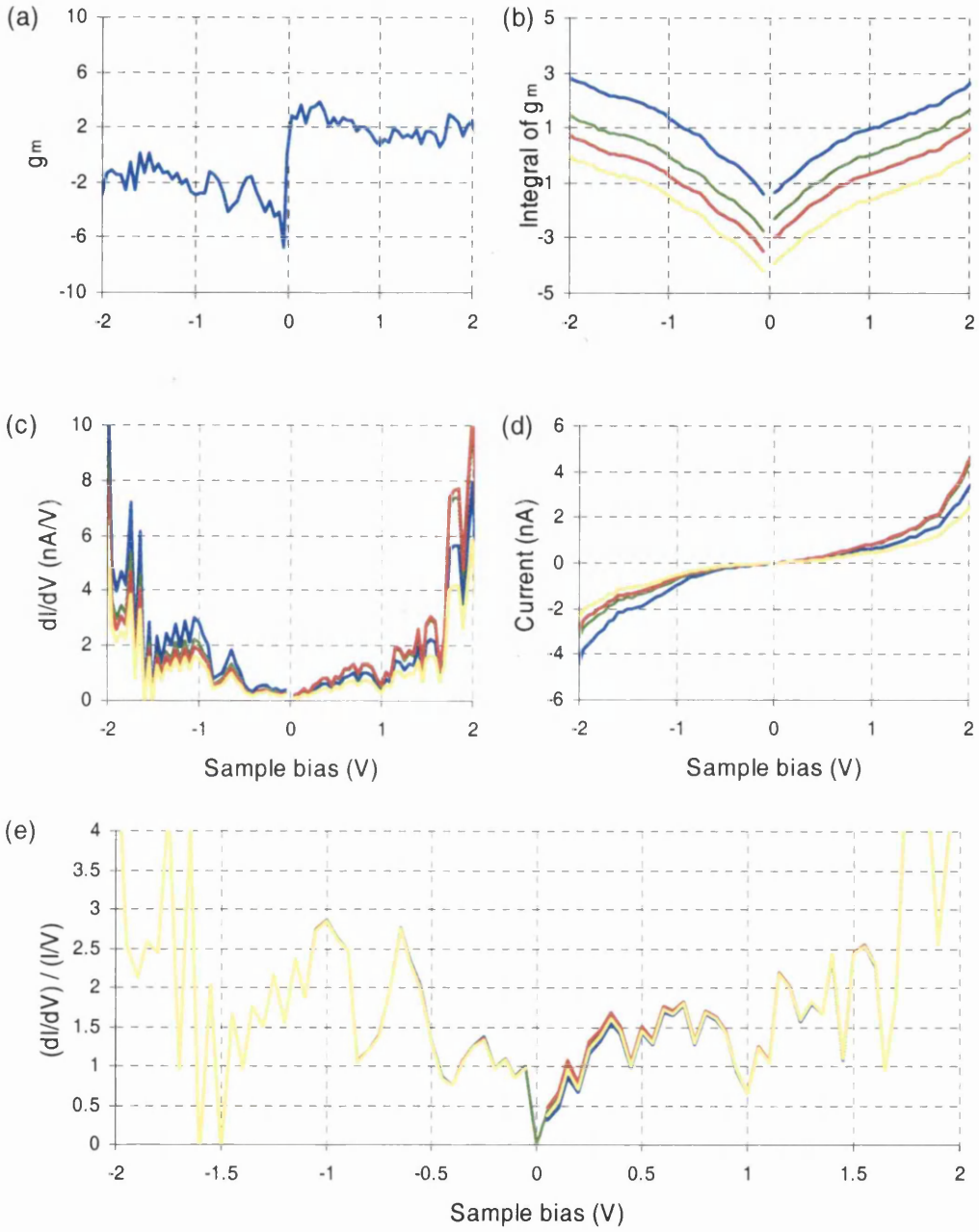


Figure 7.1.7: Curves showing quantities during the transformation of variable tip-sample separation STS data to fixed tip-sample separation equivalency. (a) Logarithmic derivative g_m, (b) the integral of g_m, (c) conductivity at equivalent fixed tip-sample separation, (d) current at equivalent fixed tip-sample and (e) normalised conductivity. Curves obtained for $|V_0|$ = 0.5V (blue), 1.0V (green), 1.5V (red) and 2.0V (yellow).

The conductivity was calculated by dividing the recorded tunnel current by the magnitude of the modulating voltage, yielding a quantity σ_m described by equation 2.12.

The analysis procedure proceeds by calculating the logarithmic derivative, g_m. This is the ratio of the conductivity σ_m to the raw current I_m as shown in equation 2.14. The procedure was based on the assumption that for small changes in separation the logarithmic derivative is approximately independent of tip-sample separation [7]. Figure 7.1.7 (a) shows g_m as a function of sample bias. The variation of the logarithmic derivative over the sample bias range indicates that it was approximately independent of tip-sample separation, verifying the assumption on which the transformation method is based. It was found that g_m was approximately independent of tip-sample separation for different locations examined within the Si(111) 7x7 unit cell.

The logarithmic derivative is integrated separately for positive and negative portions of the spectra. For each portion the integration was performed in two sections, from $|V|$ to V_0 and V_0 to 0V, where $|V|$ is the maximum sample bias at either polarity and V_0 an arbitrary value that defines the constant separation to which the conductivity and current are transformed. The magnitude of V_0 was selected to be identical for negative and positive portions of the spectra, and is chosen to lie within a region of the spectra where the current was non zero. Figure 7.1.7 (b) shows the integral of the logarithmic derivative for V_0 chosen to be: 0.5V (blue curve), 1.0V (green curve), 1.5V (red curve) and 2.0V (yellow curve). The values of V_0 selected translate to constant separations of 2.5Å, 2Å, 1.5Å and 1Å respectively as referenced to the regulated tip-sample separation.

The conductivity at a given tip-sample separation was calculated in accordance with equation 2.19 and then integrated to calculate the current at that tip-sample separation. Figure 7.1.7 (c) and (d) show conductivity and current curves for the tip-sample separations as described previously. As the tip-sample separation in reference to the regulated tip height is increased the current should increase accordingly. This trend was observed in the negative sample bias portion of figure 7.1.7 (d), but not the positive portion. For other measurements carried out at various locations within the

unit cell a deviation from this expectation was observed. This deviation from expectations was attributed to the absence of a bandgap in Si(111) 7x7, therefore the magnitude of the current at V_0 does not decrease monotonically as the V_0 approaches 0V.

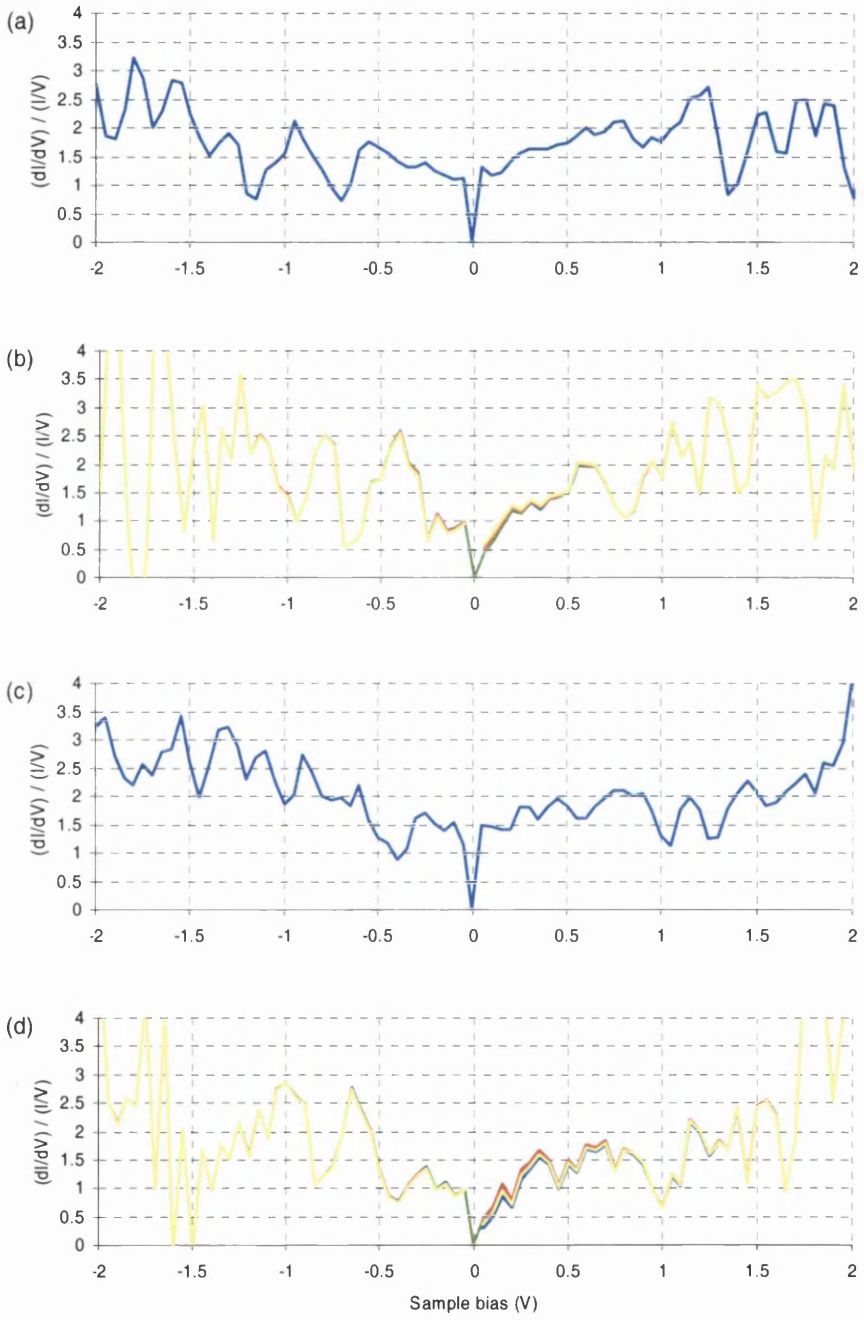


Figure 7.1.8: Normalised conductivity curves obtained from the average of several measurements made at adatom sites (graphs (a) and (b)) and corner hole sites (graphs (c) and (d)) within several Si(111) 7x7 unit cells. Graphs (a) and (c) were obtained with fixed tip-sample separation STS, whereas graphs (b) and (d) were obtained with

variable tip-sample separation STS. Graphs (b) and (d) show four curves obtained with $|V_0|$= 0.5V (Blue), 1.0V (Green), 1.5V (Red) and 2.0V (Yellow) respectively. For example the presence of surface states in the bandgap causes current peaks, which means that the current could increase as the magnitude of the sample bias decreases. The conductivity at a given tip-sample separation was normalised by the transformed current at that particular separation divided by the sample bias, in accordance with equation 2.11. Figure 7.1.7 (e) shows normalised conductivity curves for separations as previously stated. The choice of the V_0 has little effect on the normalised conductivity. Here there was no requirement to broaden or combine the normalising quantity with an offset to overcome the problem of divergence encountered with materials with a large surface band gap [8], as was the case with GaAs, since a significant current was present throughout the sample bias range.

Figure 7.1.8 shows normalised conductivity curves for the average of several spectra obtained at electronically equivalent adatom and corner hole locations within the Si(111) 7x7 unit cell. Graphs (a) and (b) were obtained at adatom sites and graphs (c) and (d) were obtained for corner hole sites using fixed (curves (a) and (c)) and variable (curves (b) and (d)) tip-sample separation methods respectively. Comparison between normalised conductivity curves obtained using fixed and variable tip-sample STS shows that curves obtained using the variable tip-sample method reveal more information in the sample bias range –1V to +1V. For example curve (b) shows well defined peaks at ~-0.8V and ~-0.4V that arise from rest atoms and adatoms respectively as discussed earlier.

7.2 Application of STM and fixed and variable tip-sample separation STS to p-type GaAs(110).

7.2.1 Preparation of GaAs (110) by cleavage.

To prepare a GaAs(110) sample for STM investigation, the sample was cut from a 3" diameter wafer of p-type GaAs substrate doped with Zn at a concentration of ~ 1 - 2.3 x 10^{18} cm^{-3}. A diamond scriber was used to scribe a 7mm wide strip from the wafer and then a fracture induced along the scribe. The process was then repeated to form a 7mm x 3mm sample which was then mounted on a custom designed sample plate.

The sample was then notched parallel to the (110) plane slightly above the clamps of the sample plate to ensure cleavage was induced in the correct position. Figure 7.2.1 shows a photograph of the sample plate. Further information regarding the sample plate design is provided by Teng [9]

Figure 7.2.1: A photograph showing the sample plate complete with sample.

The sample holder was admitted to the ESCA via a load lock and transferred to a manipulator in the evaporation chamber. Cleaving was performed at a base pressure < $1x10^{-10}$ mbar by gently applying pressure with a wobble stick to the epilayer side of the sample above the notch. After cleaving the sample was immediately transferred to the STM chamber.

7.2.2 STM of p-type GaAs(110).

Research on III-V semiconductors has been directed predominantly toward GaAs(110), due to its simple preparation by cleaving. GaAs cleaves along the (110) plane to reveal a non-polar surface, so termed because it contains equal numbers of Ga and As atoms. Half filled dangling bonds associated with Ga and As atoms are created during the cleaving process. To minimise free energy the surface undergoes relaxation and charge transfer occurs between Ga and As dangling bonds, resulting in an empty dangling bond state associated with Ga atoms and filled dangling bond state with As atoms. This causes the surface to buckle, As atoms protrude from the surface and Ga move inwards. Additionally the surface bandgap introduced by Ga and As dangling bonds is 2.4eV, which is significantly larger than the 1.4eV bulk bandgap of GaAs [10].

As is common for relaxed (110) surfaces selective imaging may be performed of Ga and As atoms by changing the sample bias [11]. With positive sample bias, electrons tunnel from the tip to empty states associated with empty Ga dangling bonds, resulting in Ga atoms appearing brighter in STM images under positive bias. Under negative sample bias electrons tunnel preferentially from filled states associated with As dangling bonds to the tip, therefore As atoms appear brighter in STM images with negative sample bias.

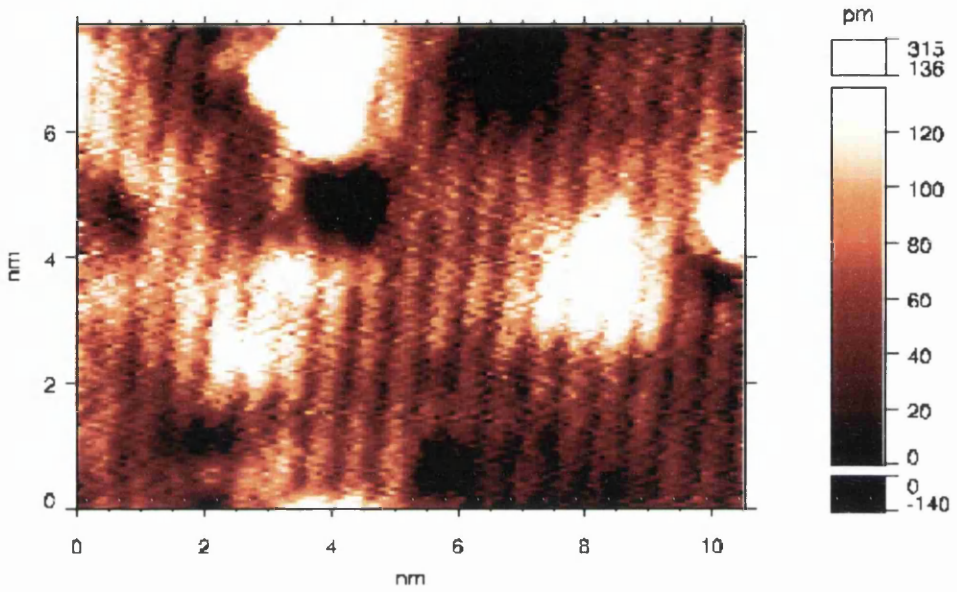


Figure 7.2.2: An STM image of the surface of p-type GaAs(110) obtained with 2.0V sample bias at a constant tunnel current of 0.3nA. The parallel corrugations visible are chains of Ga atoms.

Figure 7.2.2 shows an STM image of the clean surface of p-type GaAs(110) obtained with positive sample bias. Under these sample bias conditions tunnelling occurs from the empty dangling bond states concentrated on Ga sites to the tip. Parallel chains measured to be ~5 Å apart composed of Ga atoms are clearly visible, although individual atoms are not resolved. Figure 7.2.3 shows an STM image for the same sample obtained with negative sample bias. Tunnelling now occurs from the filled dangling bond states concentrated on As sites, and parallel chains composed of As atoms are visible. Bright and dark features observed in figure 7.2.2 are attributed to defects and dopant atoms on or near the surface.

Figure 7.2.4 shows a section of an STM image acquired with negative bias and line profiles in the [001] and [1$\bar{1}$0] directions. Measurement of the separation between

maxima denoting the positions of As atoms yields a difference of ~5.6Å between the peaks labelled 1 and 2 and ~4Å between the peaks labelled 3 and 4. This gives the unit cell marked in blue approximate dimensions 5.6 x 4Å, which are close to the theoretical dimensions of 5.65 x 3.998 Å [12].

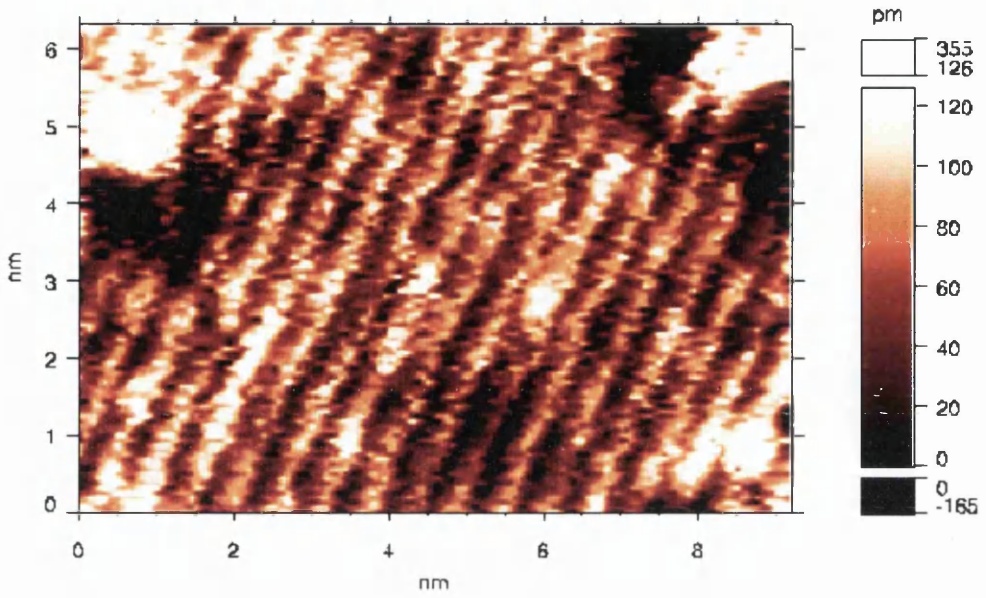


Figure 7.2.3: An STM image of the surface of p-type GaAs(110) obtained with -1.8V sample bias at a constant tunnel current of 0.6nA. The parallel corrugations visible are chains of As atoms.

7.2.3 Fixed and variable tip-sample separation STS measurements on p-type GaAs(110).

Fixed and variable tip-sample separation STS measurements on GaAs(110) were performed using a modified version of the macro described in section 7.1.3. The macro was identical in all but one respect; the linear tip-sample separation contour was generated within the macro as opposed to with external hardware. The structure of the file produced containing raw data was unaltered by the modification to the macro.

Figure 7.2.5 (a) shows raw current tunnelling spectra for fixed (blue curve) and variable (green curve) tip-sample separation STS measurements averaged for several curves, obtained on defect free areas of the sample. The magnitude of the raw current in the curve obtained using variable tip-sample separation was larger than in the curve obtained using fixed tip-sample separation at comparative sample biases. Curves (b) and (d) in figure 7.2.5 show conductivity and normalised conductivity calculated for

the fixed tip-sample separation spectra. To prevent divergence of the normalised conductivity at the bandgap edges, the total conductivity used as the normalising quantity was combined with a constant as described by equation 2.20. This method is described in chapter 2 and was necessary here because the bandgap was greater than 0.5eV [8], [13]. Combining the total conductance with a constant in the range 0.02 - 0.05 proved effective. Constant values > 0.05 removed detail from the normalised conductivity curve and values < 0.02 were insufficient to attenuate the effect of noise within the bandgap region. In figure 7.2.5 (d) choice of 0.02 as the constant value proved sufficient to reduce the noise level within the bandgap, although the effects of noise are still visible.

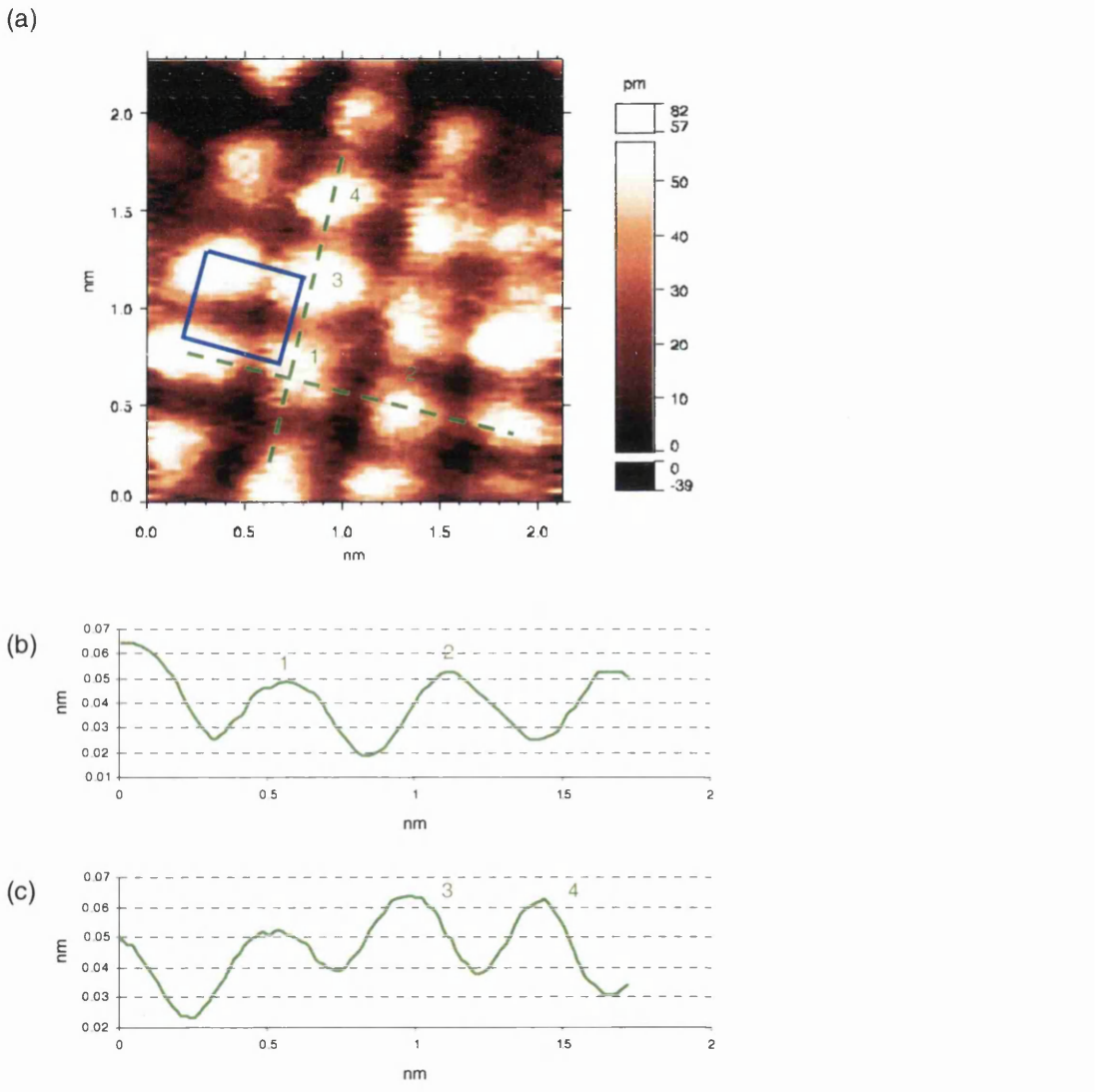


Figure 7.2.4: (a) An STM image of the surface of p-type GaAs(110) with unit cell marked in blue, obtained with -1.8V sample bias at a tunnel current of 0.4nA. (b) A line profile taken across parallel chains of As atoms in the [001] direction. (c) A line profile taken along the chains of As atoms in the [1$\bar{1}$0] direction.

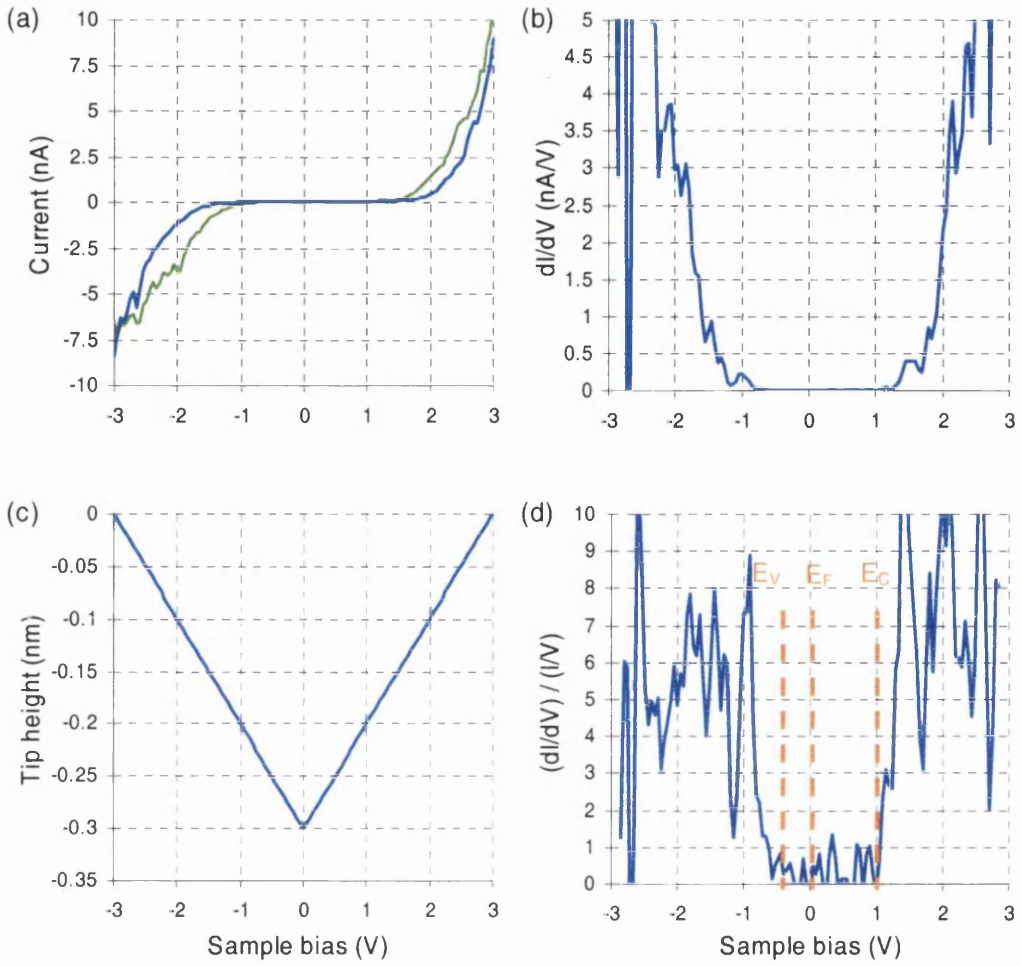


Figure 7.2.5: (a) Raw tunnel current data obtained for fixed (blue curve) and variable (green curve) tip-sample separation STS measurements. (b) Conductivity calculated from the raw tunnel current for fixed tip-sample separation. (c) Tip-sample separation in reference to the regulated tip height for variable tip-sample separation measurements. (d) Normalised conductivity for fixed tip-sample separation measurements with E_V denoting the valence band maximum, E_C the conduction band minimum and E_F the Fermi level.

E_V and E_C denote the position of the valence and conduction band edges and E_F the position of the Fermi level within the normalised conductivity curve. The valence band edge is positioned at –0.4V and the conduction band edge at 1V indicating a bandgap of ~1.4eV which is characteristic of GaAs. The Fermi level, E_F is positioned close to the valence band edge indicating p-type material. The linear tip-sample separation contour, figure 7.2.5 (c) has a gradient of 1Å/V and attains a peak displacement from the regulated tip height of 3Å. As was observed with Si, increasing

this gradient provides greater amplification of the tunnel current and decreasing it a corresponding decrease.

Raw data obtained using variable tip-sample separation STS measurements was processed using the same procedure as described in detail in section 7.1.3, therefore discussion is confined here to the effect of parameters used in the procedure and the results themselves.

The logarithmic derivative, figure 7.2.6 (a) was approximately independent of tip sample separation for positive sample bias and also negative sample bias until ~-0.2V where a discontinuity occurs. The effect of the discontinuity propagates through the transformation procedure, appearing as a spike in the normalised conductance curve, figure 7.2.6 (e).

The choice of V_0 and therefore the separation to which the conductivity and current are transformed can be seen in figure 7.2.6 (c) and (d). As $|V_0|$ was reduced and therefore the separation from the regulated tip height increased, the amplitude of the conductivity and thus the current increase. Features in the conductivity spectra remain in the same spectral locations, but increase in scale. The choice of V_0 has no obvious effect on the normalised conductivity, figure 7.2.6 (e), since the curves for different values of V_0 collapse into a single curve.

It was unnecessary to combine the total conductance with an offset in order to prevent the normalised conductivity diverging at the band edges, since a current due to surface states was present within the bandgap. The conduction band onset, E_C was located at ~0.9V and is marked by a rapid increase in the density of states. Whereas the valence band onset, E_V was located at ~-0.4V and marked by a gradual increase in the density of states. Therefore giving a bandgap of ~1.3eV which was slightly smaller than the 1.4eV characteristic of GaAs.

In figure 7.2.6 (e) an additional feature was observed within the bulk bandgap, situated close to the conduction band onset. This feature marked by a 'D' indicates the presence of empty surface states. In work carried out by Feenstra [14] on clean III-V semiconductor surfaces using variable tip-sample separation STS, he makes a similar

observation in measurements made on p-type GaAs. A bump was observed close to the conduction band onset, which he attributes to dopants. It seems likely that the component D indicates a dopant induced feature.

A comparison between normalised conductivity curves obtained using fixed and variable tip-sample separation STS, figures 7.2.5 and 7.2.6 respectively, shows that in the latter the band edges are better defined and surface states previously veiled in noise are clearly visible.

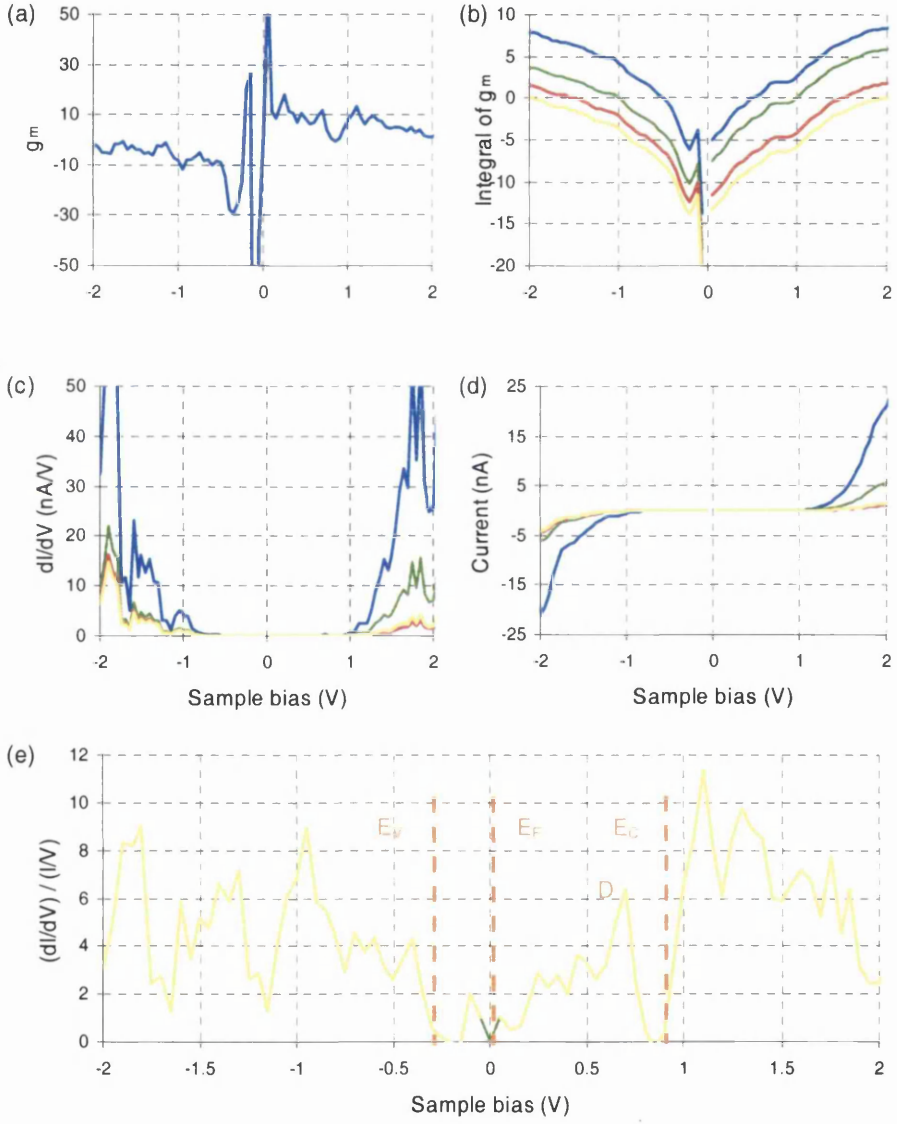


Figure 7.2.6: Curves showing quantities during the transformation of variable tip-sample separation STS data to fixed tip-sample separation equivalency. (a) Logarithmic derivative g_m, (b) the integral of g_m, (c) conductivity at equivalent fixed tip-sample separation, (d) current at equivalent fixed tip-sample and (e) normalised conductivity. Curves obtained for $|V_0|$ = 0.5V (blue), 1.0V (green), 1.5V (red) and 2.0V (yellow).

7.3 References

[1] G.Binning, H. Rohrer, Ch. Gerber, and E.Weibel, Phys. Rev. Lett, **50**, 120 (1983).

[2] K.Takayanagi, Y. Tanishiro, M. Takahashi, and S. Takahashi, J. Vac. Sci. Technol, A **3**, 1502 (1985).

[3] R.J. Hamers, R.M. Tromp, and J.E. Demuth, Phys. Rev. Lett., **56**, 1972 (1986).

[4] R.J. Hamers, J. Vac. Sci. Techno, B **6**, 1462 (1988).

[5] R.J. Hamers, D.F. Padowitz, 'Scanning Probe Microscopy and Spectroscopy', Chapter 4, **79-83**, Edit D.A. Bonnel, Second edition, Wiley-VCH (2001).

[6] Omicron NanoTechnology, 'The SCALA PRO Software Manual', Version 4.1 (2001).

[7] R.M. Feenstra, J.A. Stroscio, 'Methods of Experimental Physics: Scanning Tunnelling Microscopy', Chapter 4, **141**, Edit J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[8] R.M. Feenstra, J.A. Stroscio, 'Methods of Experimental Physics: Scanning Tunnelling Microscopy', Chapter 4, **118**, Edit J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[9] K.S. Teng, PhD thesis, University of Wales Swansea, (2000).

[10] R.M. Feensra, J.A. Stroscio, 'Methods of Experimental Physics: Scanning Tunnelling Microscopy', Chapter 5, **251-258**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[11] R.M. Feenstra, J.A. Stroscio, J. Tersoff, and A.P. Fein, Phys. Rev. Lett., **58**, 1192 (1987).

[12] A.R. Lubinsky, C.B. Duke, B.W. Lee, and P. Mark, Phys. Rev. Lett., **36**, 1058 (1976).

[13] M. Prietsch, A. Samsavar, and R. Ludeke, Phys. Rev. B, **43**, 11850 (1991).

[14] R.M. Feenstra, Phys. Rev. B, **50**, 4561 (1994).

Chapter 8

Conclusions and Future Work

8.1 Conclusions and Future work.

The work presented in this thesis was principally concerned with the design, development and verification of instrumentation to operate in conjunction with Omicron Micro-STM and STM/SEM-HC Scanning Tunnelling Microscope systems, to implement variable tip-sample separation STS.

Typically, STS measurements are performed concurrently with STM imaging, to investigate the localised electronic properties of features on an atomic scale, for a diverse range of materials. When STS is performed concurrently with STM, detailed information about the nature of electronic states associated with particular surface features can be deduced. Conventionally, STS measurements are performed with the tip-sample separation fixed throughout the period of the measurement. The variable tip-sample separation technique overcomes problems associated with fixed tip-sample separation by varying the tip-sample separation in a controlled manner during the measurement, to increase the dynamic range. To accomplish this, instrumentation and software were developed to generate both linear and non-linear tip-sample separation contours during variable tip-sample separation STS.

Evaluation of the instrumentation and variable tip-sample separation technique was performed by completing fixed and variable tip-sample separation STS on p-type Si(111) 7x7 and p-type GaAs(110), and comparing the results obtained in each case with previous work and theoretical expectations.

In order to generate the tip-sample separation contours during variable tip-sample separation STS, solutions based on hardware and software were successfully developed and tested. In the case of the non-linear tip-sample contour only a partial hardware solution was realised, due to problems encountered with conductivity measurements. To fully realise the non-linear solution, additional work is required to develop circuitry to deduce the next step size from the sample conductivity. This development relies on successful resolution of the problems associated with conductivity measurements, which are discussed later.

The variable tip-sample separation technique requires measurement of sample conductivity for every data point within a spectra [1]. To accomplish this, two methods were investigated. The first method was implemented directly in the spectroscopy macros and the second using a Lock-in amplifier.

The first method of conductivity measurement was implemented entirely within spectroscopy macros. This method approximates a sinusoidal modulation of the sample bias by subtracting and then adding an offset to the sample bias, performing a tunnel current measurement at each offset. In practice an offset in the range 10 – 50mV generally proved acceptable. The offset depended on the sample conductivity and varied between samples. The advantages of this method were short measurement times and ease of implementation.

The second method, using the Lock-in amplifier proved unsuccessful and was abandoned in favour of the first method. When conductivity measurements were performed with the Lock-in, the conductivity spectra obtained contained artefacts which were introduced by the Lock-in. These artefacts were related to the time the modulation was present for during each data point. As the period of time the reference signal was present for was increased, the amplitude of the artefacts in the conductivity spectra reduced. However, this increased the measurement time for the complete spectra. The increase in measurement time increased the sensitivity of the measurement to thermal drift, introducing uncertainties as to the physical location of the measurement on the sample surface. In an effort to reduce the measurement time, the number of data points in the spectra was reduced. This however reduced the resolution without significantly reducing the overall measurement time.

It was observed that as the frequency of the reference signal was increased, there was a corresponding increase in the tunnel current component caused by the tip-sample capacitance. This capacitive component manifests itself as an increasing offset in the conductivity spectra. It is probable that the tip-sample separation capacitance perturbs the conductivity measurement; thus the conductivity spectra include artefacts introduced by the capacitance.

To improve the accuracy of the first method and to investigate problems of long measurement times, and capacitance induced effects associated with the second method, future work is necessary. Investigation of the source(s) of the capacitance and methods of correcting for its effects would be a suitable starting point.

Study of p-type Si(111) 7x7 was undertaken to appraise the variable tip-sample separation technique. Samples of p-type Si(111) were prepared from a B doped Si wafer and processed in vacuum to remove the native oxide layer and contamination. Samples were imaged using constant current mode STM for both positive and negative sample bias. STM images obtained showed the characteristic unit cell of the Si(111) 7x7 reconstruction, with 4 deep depressions at the corners and 12 protrusions, clearly marking the position of corner holes and adatoms respectively [2].

Under positive sample bias conditions the protrusions marking the positions of adatoms in the unit cell appear of uniform brightness, indicating similar empty state density. Whereas, under negative sample bias conditions adatoms within the faulted half of 7x7 unit cell appear brighter than their counterparts in the un-faulted half, indicating a greater filled state density in the faulted half. Additionally, the three adatoms closest to the corner holes appear brighter than their counterparts, possessing the greatest density of filled states.

Study of p-type GaAs(110) was undertaken in order to evaluate the variable-tip sample separation technique in a material system possessing a band gap. Samples of p-type GaAs(110) were prepared form a Zn doped GaAs wafer and cleaved in vacuum to reveal the non-polar (110) surface.

The freshly cleaved sample surface was imaged using constant current mode STM, for both positive and negative sample bias. STM images of the GaAs(110) surface obtained at positive sample bias show parallel lines of regularly spaced protrusions, denoting the position of Ga atoms. Whereas, images obtained at negative sample bias show similar parallel lines of regularly spaced protrusions, where now the protrusions mark the position of the As atoms. The dimensions of the unit cell are measured from

these images to be 5.6 x 4Å, which are in good agreement with the theoretical dimensions of 5.65 x 3.998Å [3].

Variable and fixed tip-sample separation spectra were obtained consecutively from well ordered regions of both the Si(111) 7x7 and GaAs(110) surfaces. Several spectra acquired from electronically equivalent sites were averaged and then mathematically processed using a custom designed Microsoft Excel spreadsheet. The spreadsheet implements the mathematical transformation method suggested by Feenstra *et al* [4], [5] to process variable tip-sample separation spectra.

The magnitude of the raw current, at comparative sample biases for variable tip-sample separation in both Si(111) and GaAs(110) raw current spectra is greater than that obtained for fixed tip-sample separation. This is consistent with theory which predicts an increase in tunnel current as the tip-sample separation is reduced [6]. For both materials, as the slope of the tip-sample separation contour was increased a corresponding increase in the amplification in the raw current was observed. Similarly, a reduction in the slope reduced the amplification.

The assumption on which the transformation method Feenstra proposes is based, states that the logarithmic derivative (equation 2.14) is approximately independent of tip-sample separation for small changes in separation [5]. This assumption was verified for both Si(111) and GaAs(110) samples, validating the transformation method. The choice of tip-sample separation to which the spectra acquired using the variable tip-sample separation method are transformed, proved to have no obvious effect on the normalised conductivity spectra obtained.

The normalised conductivity spectra calculated from fixed and variable tip-sample separation STS measurements, performed at adatom and corner hole sites within the Si(111) 7x7 unit cell, show surface state features throughout the bulk bandgap region. A comparison between spectra obtained with fixed and variable tip-sample separation shows that spectral features are better defined in the latter, and that the positions of maximums in surface state density is different for adatom and corner hole sites. The different energetic positions of surface state features illustrates the difference in electronic properties between adatom and corner hole sites.

The normalised conductivity spectra calculated from fixed and variable tip-sample separation STS measurements performed on GaAs(110) both show a bandgap of ~1.4eV, which is characteristic of bulk GaAs. Additionally, in both spectra the Fermi level is positioned close to the valence band edge, indicating p-type material. The magnitude of the noise in the bandgap region of the normalised conductivity spectra obtained using the variable tip-sample separation technique is much less than in the normalised conductivity spectra employing fixed tip-sample separation. This reduction in noise magnitude within the bandgap region allowed previously unresolved surface states to be clearly resolved. The surface states evidence themselves as a feature close to the conduction band onset, and are most likely introduced by dopants. This observation of surface states is similar to observations made by Feenstra [7].

In both material systems investigated here, the variable tip-sample separation technique enhances definition of the spectral features and signal to noise ratio in STS measurements. In future work, an investigation is required to ascertain the sensitivity of the measurement to tip induced effects and the resulting effect to spectra obtained.

8.2 References.

[1] J.A. Stroscio, R.M. Feenstra, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 4, **135-138**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[2] K.Takayanagi, Y. Tanishiro, M. Takahashi and S. Takahashi, J. Vac. Sci Technol, A **3**, 1502 (1985).

[3] A.R. Lubinsky, C.B. Duke, B.W. Lee, and P. Mark, Phys. Rev. Lett., **36**, 1058 (1976).

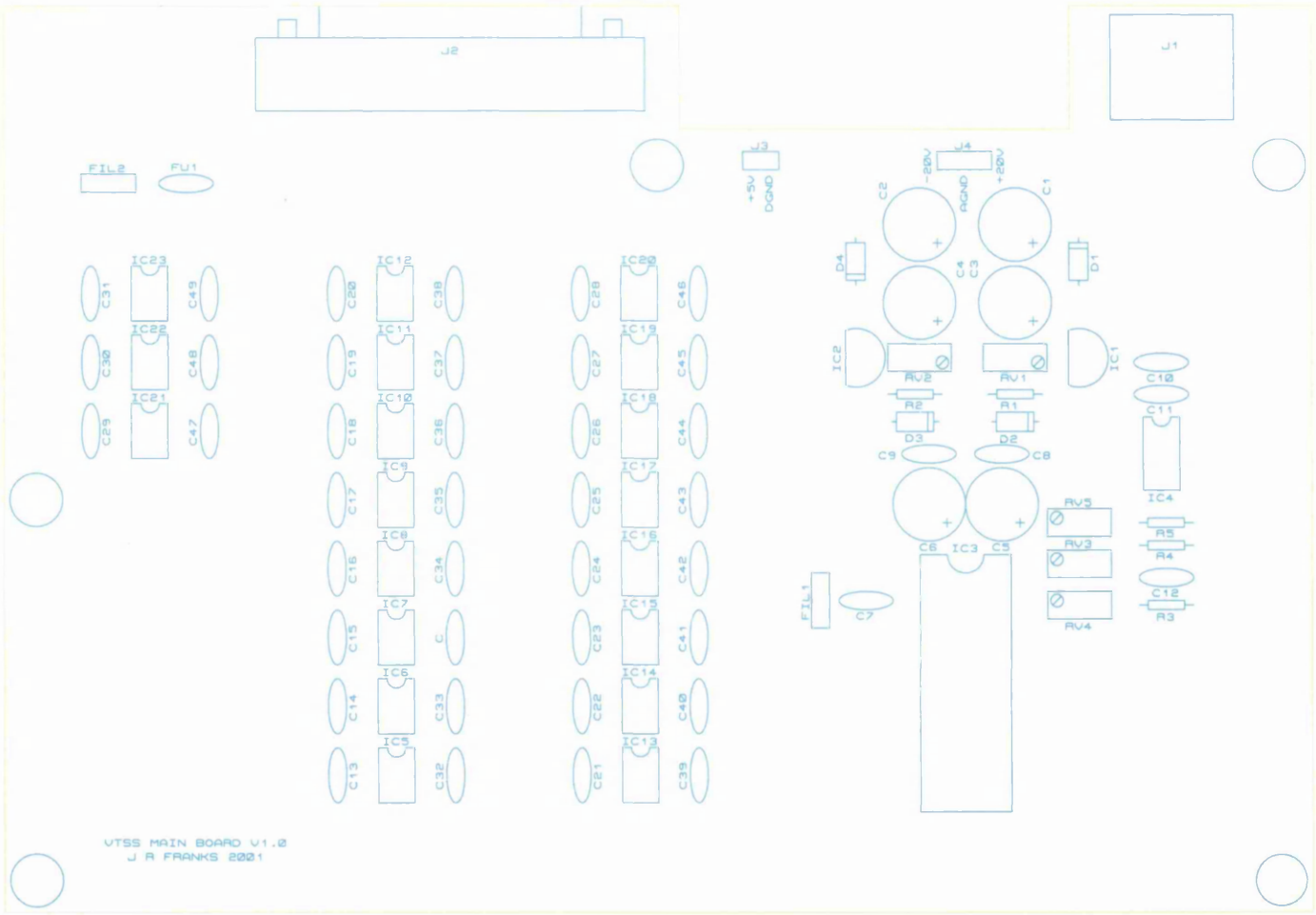
[4] C.K. Shih, R.M. Feenstra and G.V.Chandrashekhar, Phys. Rev. B **43**, 7913 (1991).

[5] J.A. Stroscio, R.M. Feenstra, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 4, **141-145**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

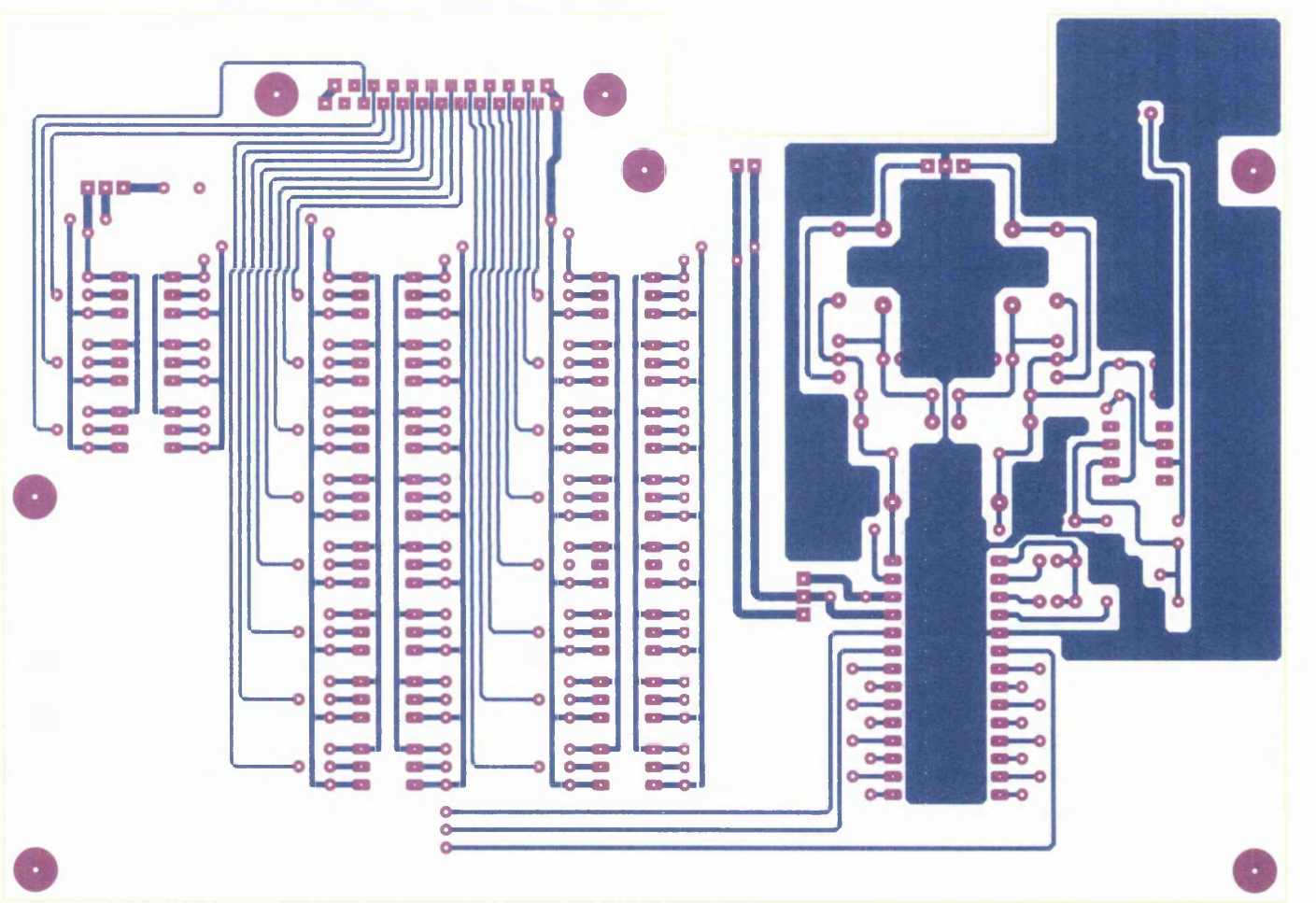
[6] J. Tersoff, N.D. Lang, 'Methods of Experimental Physics: Scanning Tunneling Microscopy', Chapter 1, **1 – 3**, Edit. J.A. Stroscio and W.J. Kaiser, Academic Press (1993).

[7] R.M. Feenstra, Phys. Rev. B, **50**, 4561 (1994).

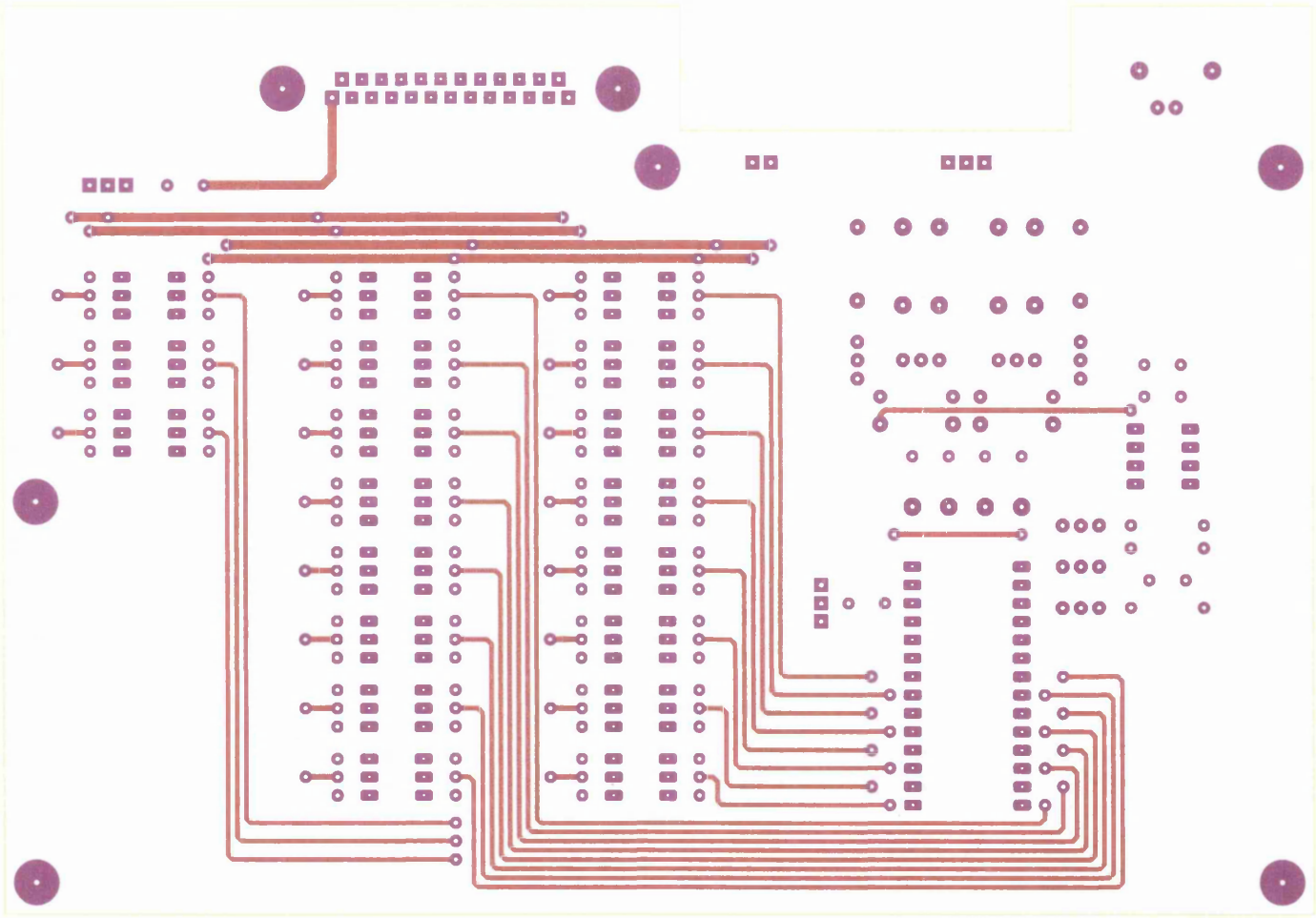
Appendix A



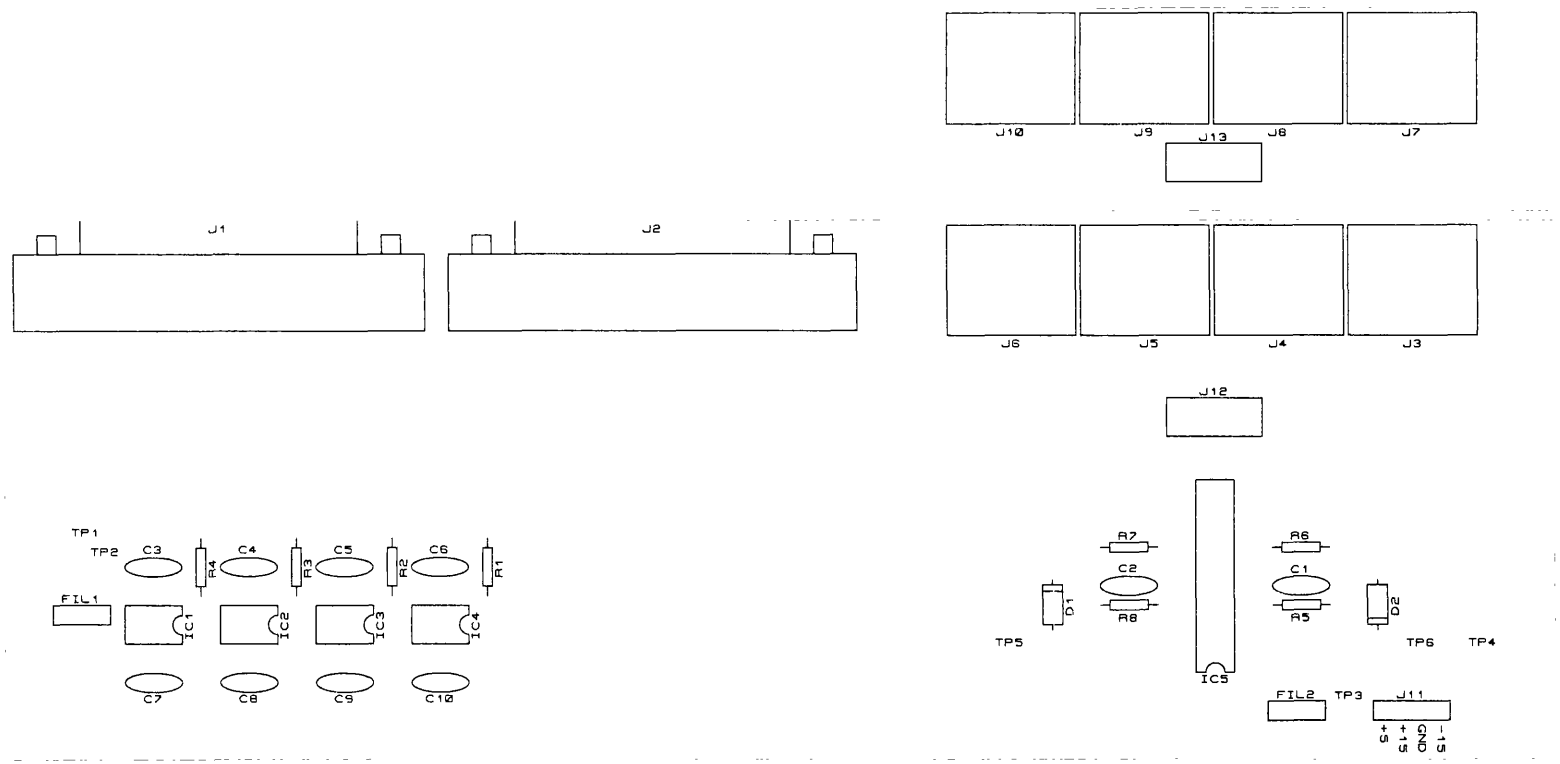
Appendix A1 - Printed circuit artwork for the digital to analog converter (DAC) module.



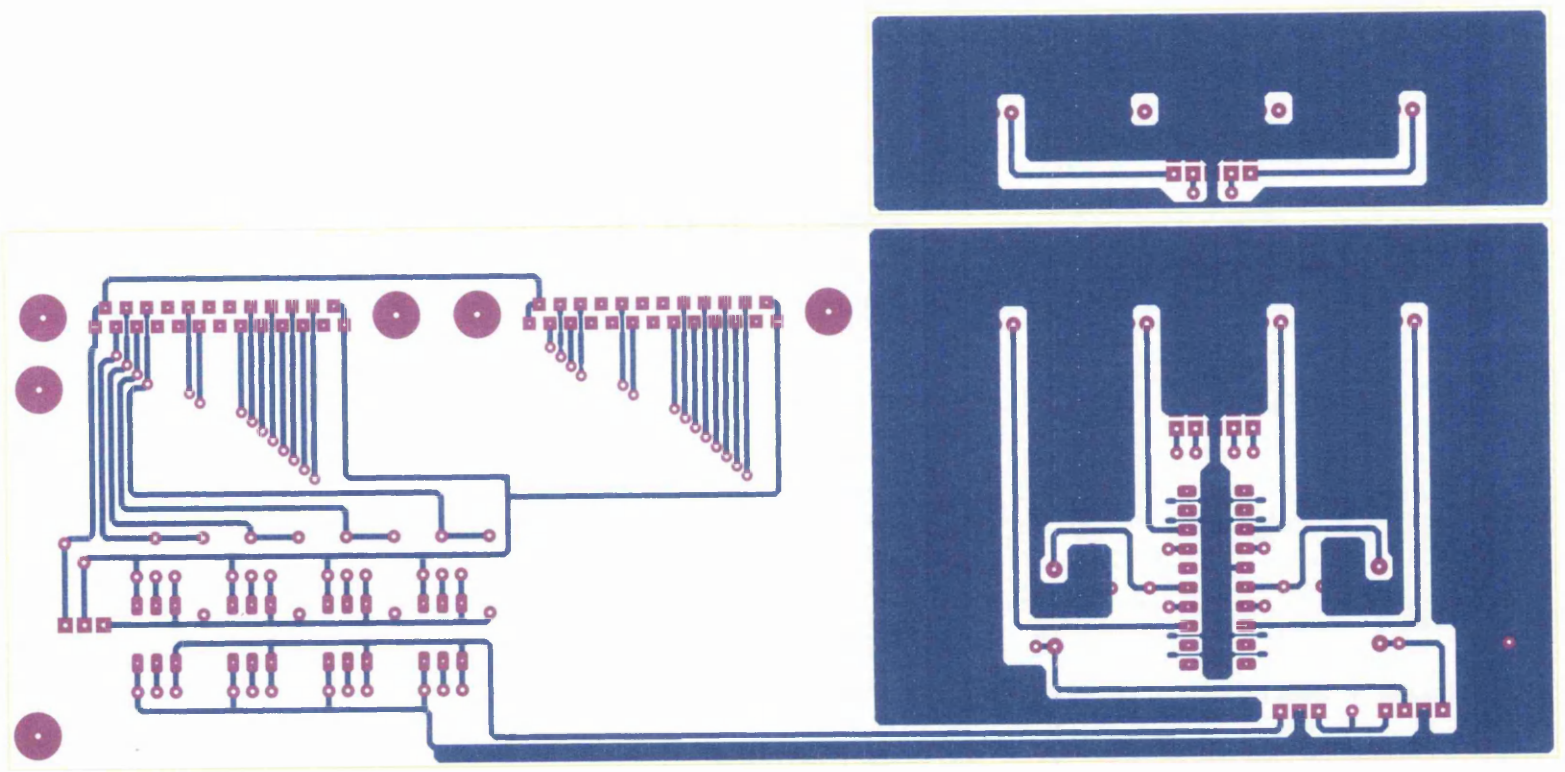
Appendix A1 - Printed circuit artwork for the digital to analog converter (DAC) module.



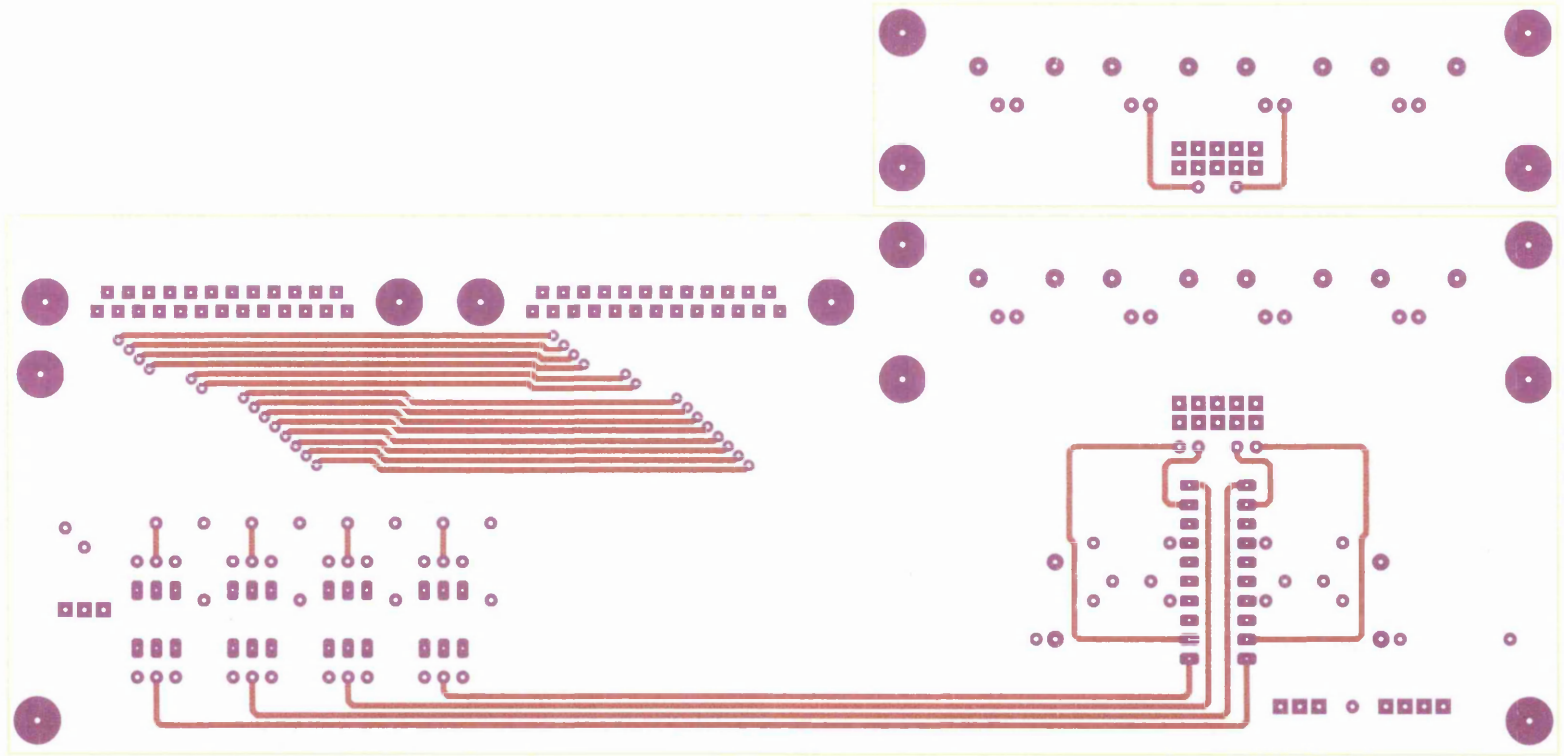
Appendix A1 - Printed circuit artwork for the digital to analog converter (DAC) module.



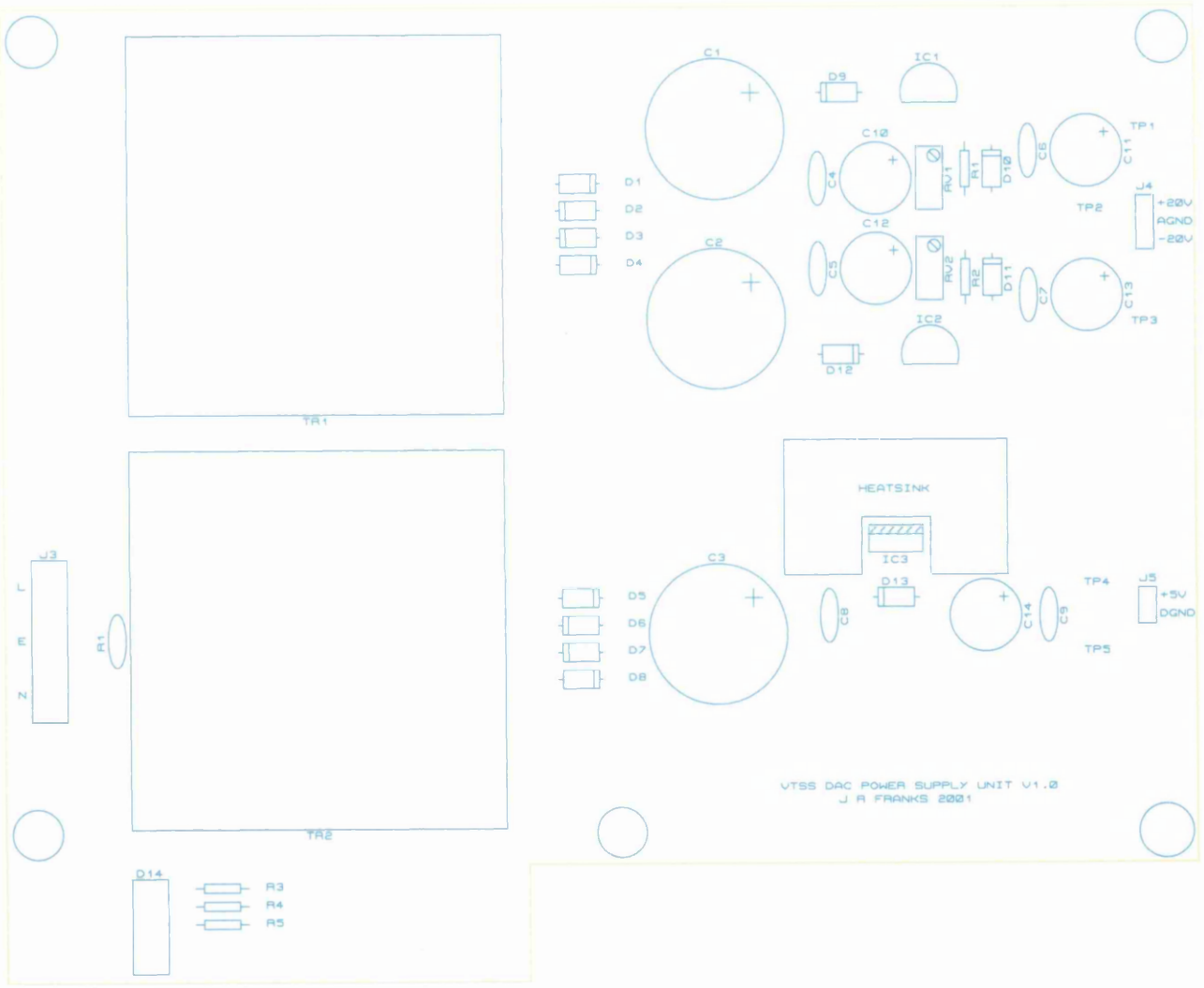
Appendix A2 - Printed circuit artwork for the signal routing module.



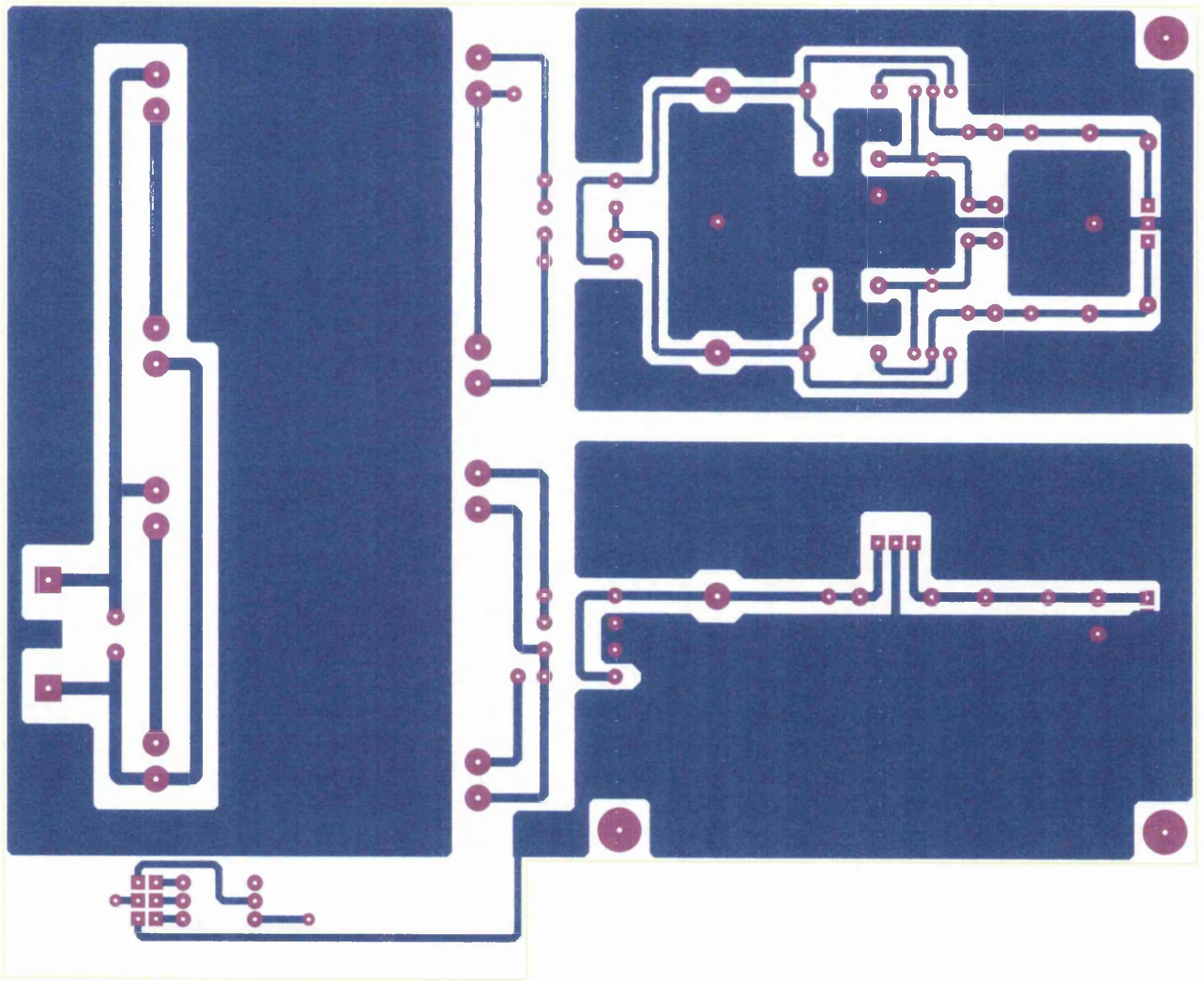
Appendix A2 - Printed circuit artwork for the signal routing module.



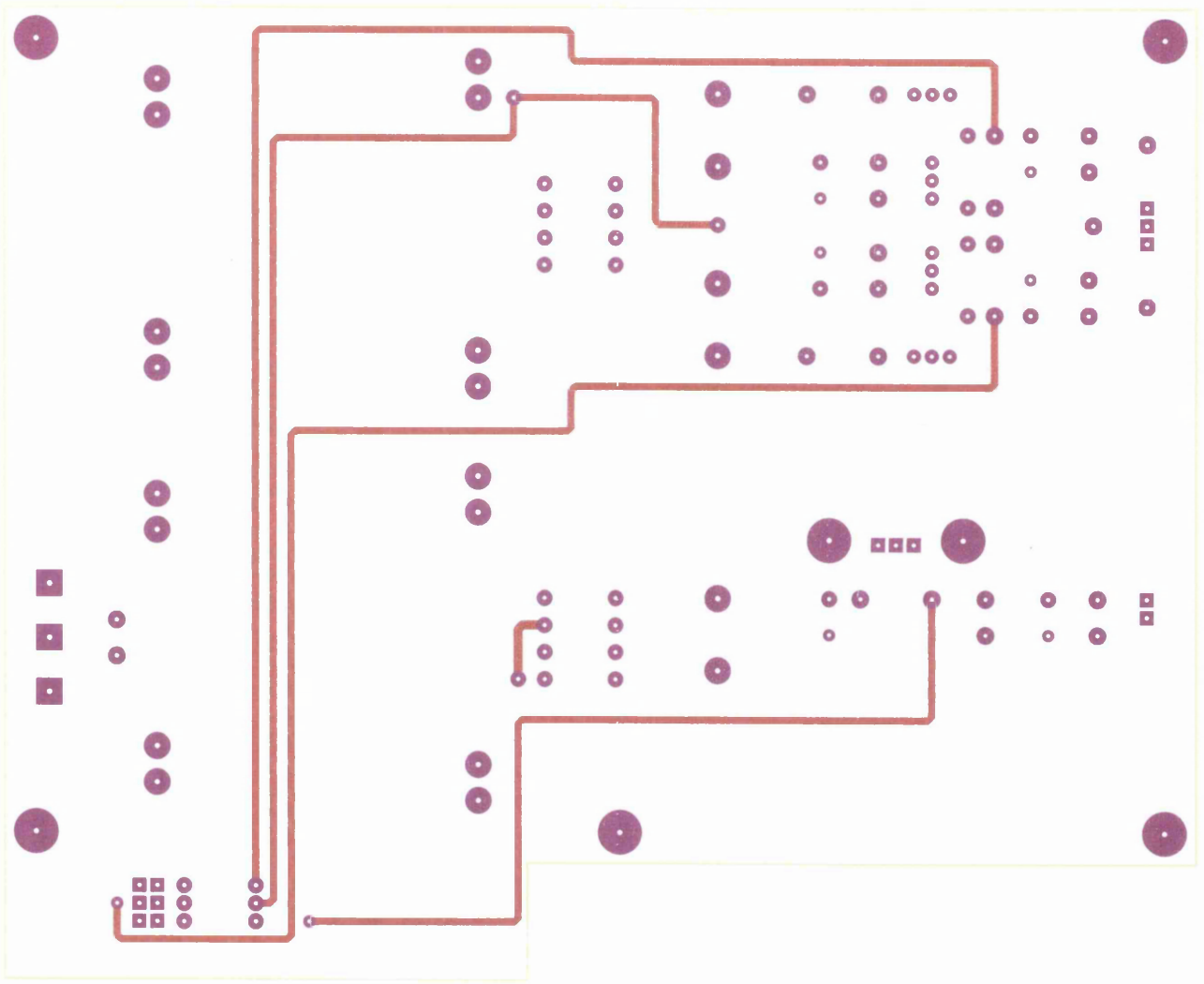
Appendix A2 - Printed circuit artwork for the signal routing module.



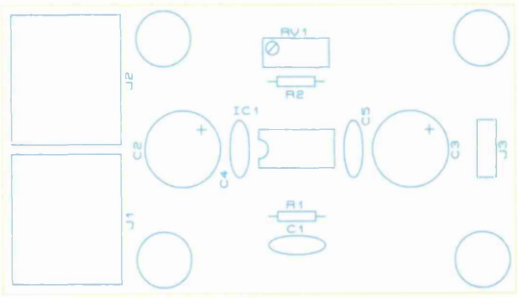
Appendix A3 - Printed circuit artwork for the power supply.



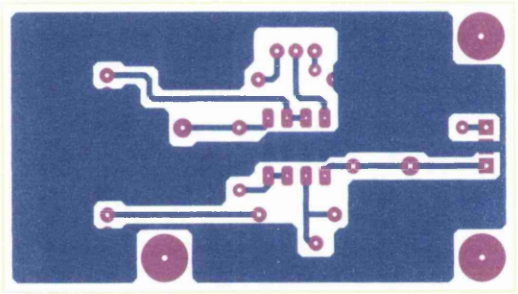
Appendix A3 - Printed circuit artwork for the power supply.



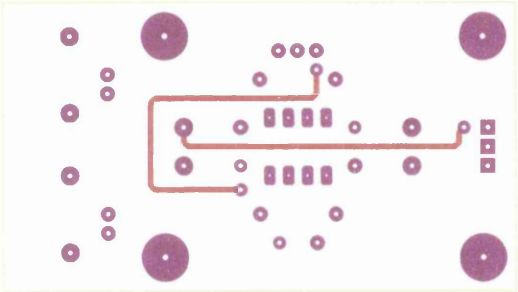
Appendix A3 - Printed circuit artwork for the power supply.



Appendix A4 - Printed circuit artwork for the attenuator module.



Appendix A4 - Printed circuit artwork for the attenuator module.



Appendix A4 - Printed circuit artwork for the attenuator module.

Appendix B

Appendix B1 – Microcontroller Code.

```
*****************************************************************************************
***  Title:     VTSS main processor                                                   ***
***  File name: VTSS2.ASM                                                             ***
***  Processor: MC68HC912B32                                                          ***
***  Author:    J R Franks                                                            ***
***  Date:      16th April 2002                                                       ***
***  Function:  Calculates and applies linear S-Contour to Z-offset input             ***
***             during spectroscopy therfore enabling VTSS.                           ***
***  Updated:   30th May 2002 Addition of non-linear program completed                ***
***             3rd July 2002 Removal of SPI routines and addition of SCI routines    ***
***             and fire renamed                                                      ***
*****************************************************************************************

*****************************************************************************************
*** Register equates                                                                    ***
*****************************************************************************************

PORTA           EQU     $0000               ;Port A data register
DDRA            EQU     $0002               ;Port A data direction register
PORTB           EQU     $0001               ;Port B data register
DDRB            EQU     $0003               ;Port B data direction register
PORTE           EQU     $0008               ;Port E data register
DDRE            EQU     $0009               ;Port E data direction register
PEAR            EQU     $000A               ;Port E assignment register
MODE            EQU     $000B               ;Mode rgister
PUCR            EQU     $000C               ;Pull up control register
RDRIV           EQU     $000D               ;Reduced drive register
INIRAM          EQU     $0010               ;RAM initialization register
INIREG          EQU     $0011               ;Register initialization register
INIEE           EQU     $0012               ;EEPROM initialization register
MISC            EQU     $0013               ;Misellaneous mapping control
RTICTL          EQU     $0014               ;Real-time interrupt control
RTIFLG          EQU     $0015               ;Real-time interrupt flag
COPCTL          EQU     $0016               ;COP control register
COPRST          EQU     $0017               ;Arm/reset COP timer
INTCR           EQU     $001E               ;Interrupt control register
HPRIO           EQU     $001F               ;Highest priority I interrupt
BRKCT0          EQU     $0020               ;Breakpoint control register 0
BRKCT1          EQU     $0021               ;Breakpoint control register 1
BRKAH           EQU     $0022               ;Breakpoint address register high
BRKAL           EQU     $0023               ;Breakpoint address register low
BRKDH           EQU     $0024               ;Breakpoint data register high
```

```
BRKDL           EQU     $0025           ;Breakpoint data register low
PWCLK           EQU     $0040           ;PWM clocks and concatenate
PWPOL           EQU     $0041           ;PWM clock select and polarity
PWEN            EQU     $0042           ;PWM enable register
PWPRES          EQU     $0043           ;PWM prescaler counter
PWSCAL0         EQU     $0044           ;PWM scale register 0
PWSCNT0         EQU     $0045           ;PWM scale counter register 0
PWSCAL1         EQU     $0046           ;PWM scale register 1
PWSCNT1         EQU     $0047           ;PWM scale counter register 1
PWCNT0          EQU     $0048           ;PWM channel counter register 0
PWCNT1          EQU     $0049           ;PWM channel counter register 1
PWCNT2          EQU     $004A           ;PWM channel counter register 2
PWCNT3          EQU     $004B           ;PWM channel counter register 3
PWPER0          EQU     $004C           ;PWM channel period register 0
PWPER1          EQU     $004D           ;PWM channel period register 1
PWPER2          EQU     $004E           ;PWM channel period register 2
PWPER3          EQU     $004F           ;PWM channel period register 3
PWDTY0          EQU     $0050           ;PWM channel duty register 0
PWDTY1          EQU     $0051           ;PWM channel duty register 1
PWDTY2          EQU     $0052           ;PWM channel duty register 2
PWDTY3          EQU     $0053           ;PWM channel duty register 3
PWCTL           EQU     $0054           ;PWM control register
PWTST           EQU     $0055           ;PWM special mode
PORTP           EQU     $0056           ;PORTP data register
DDRP            EQU     $0057           ;PORTP data direction register
ATDCTL0         EQU     $0060           ;ATD control register 0
ATDCTL1         EQU     $0061           ;ATD control register 1
ATDCTL2         EQU     $0062           ;ATD control register 2
ATDCTL3         EQU     $0063           ;ATD control register 3
ATDCTL4         EQU     $0064           ;ATD control register 4
ATDCTL5         EQU     $0065           ;ATD control register 5
ATDSTATH        EQU     $0066           ;ATD status register A
ATDSTATL        EQU     $0067           ;ATD status register B
ATDTSTH         EQU     $0068           ;ATD test register high
ATDTSTL         EQU     $0069           ;ATD test register low
PORTAD          EQU     $006F           ;PORTAD data input register
ADRx0H          EQU     $0070           ;ATD result register 0 high
ADRx0L          EQU     $0071           ;ATD result register 0 low
ADRx1H          EQU     $0072           ;ATD result register 1 high
ADRx1L          EQU     $0073           ;ATD result register 1 low
ADRx2H          EQU     $0074           ;ATD result register 2 high
ADRx2L          EQU     $0075           ;ATD result register 2 low
```

```
ADRx3H          EQU     $0076           ;ATD result register 3 high
ADRx3L          EQU     $0077           ;ATD result register 3 low
ADRx4H          EQU     $0078           ;ATD result register 4 high
ADRx4L          EQU     $0079           ;ATD result register 4 low
ADRx5H          EQU     $007A           ;ATD result register 5 high
ADRx5L          EQU     $007B           ;ATD result register 5 low
ADRx6H          EQU     $007C           ;ATD result register 6 high
ADRx6L          EQU     $007D           ;ATD result register 6 low
ADRx7H          EQU     $007E           ;ATD result register 7 high
ADRx7L          EQU     $007F           ;ATD result register 7 low
TIOS            EQU     $0080           ;Timer IC/OC select register
CFORC           EQU     $0081           ;Timer compare force register
OC7M            EQU     $0082           ;Timer output compare 7 mask
OC7D            EQU     $0083           ;Timer output compare 7 data
TCNTH           EQU     $0084           ;Timer count register high
TCNTL           EQU     $0085           ;Timer count register low
TSCR            EQU     $0086           ;Timer system control register
TCTL1           EQU     $0088           ;Timer control register 1
TCTL2           EQU     $0089           ;Timer control register 2
TCTL3           EQU     $008A           ;Timer control register 3
TCTL4           EQU     $008B           ;Timer control register 4
TMSK1           EQU     $008C           ;Timer mask register 1
TMSK2           EQU     $008D           ;Timer mask register 2
TFLG1           EQU     $008E           ;Timer interrupt flag 1
TFLG2           EQU     $008F           ;Timer interrupt flag 2
TC0H            EQU     $0090           ;Timer IC/OC 0 high register
TC0L            EQU     $0091           ;Timer IC/OC 0 low register
TC1H            EQU     $0092           ;Timer IC/OC 1 high register
TC1L            EQU     $0093           ;Timer IC/OC 1 low register
TC2H            EQU     $0094           ;Timer IC/OC 2 high register
TC2L            EQU     $0095           ;Timer IC/OC 2 low register
TC3H            EQU     $0096           ;Timer IC/OC 3 high register
TC3L            EQU     $0097           ;Timer IC/OC 3 low register
TC4H            EQU     $0098           ;Timer IC/OC 4 high register
TC4L            EQU     $0099           ;Timer IC/OC 4 low register
TC5H            EQU     $009A           ;Timer IC/OC 5 high register
TC5L            EQU     $009B           ;Timer IC/OC 5 low register
TC6H            EQU     $009C           ;Timer IC/OC 6 high register
TC6L            EQU     $009D           ;Timer IC/OC 6 low register
TC7H            EQU     $009E           ;Timer IC/OC 7 high register
TC7L            EQU     $009F           ;Timer IC/OC 7 low register
PACTL           EQU     $00A0           ;Pulse accumulator control register
```

```
PAFLG           EQU     $00A1           ;Pulse accumulator flag register
PACN3           EQU     $00A2           ;Pulse accumulator count register 3
PACN2           EQU     $00A3           ;Pulse accumulator count register 2
PACN1           EQU     $00A4           ;Pulse accumulator count register 1
PACN0           EQU     $00A5           ;Pulse accumulator count register 0
MCCTL           EQU     $00A6           ;16-bit moudulus down-counter
                                        ;control register
MCFLG           EQU     $00A7           ;16-bit modulus down_counter control
                                        ;register
ICPACR          EQU     $00A8           ;Input control pulse accumulators
                                        ;control register
DLYCT           EQU     $00A9           ;Delay counter control register
ICOVW           EQU     $00AA           ;Input control overwrite register
ICSYS           EQU     $00AB           ;Input control system control
                                        ;register
TIMTST          EQU     $00AD           ;Timer test register
PORTT           EQU     $00AE           ;PORTT data register
DDRT            EQU     $00AF           ;PORTT data direction register
PBCTL           EQU     $00B0           ;16-bit pulse accumulator B control
                                        ;register
PBFLG           EQU     $00B1           ;Pulse accumulator B flag register
PA3H            EQU     $00B2           ;8-bit pulse accumulator holding
                                        ;register 3
PA2H            EQU     $00B3           ;8-bit pulse accumulator holding
                                        ;register 2
PA1H            EQU     $00B4           ;8-bit pulse accumulator holding
                                        ;register 1
PA0H            EQU     $00B5           ;8-bit pulse accumulator holding
                                        ;register 0
MCCNT0          EQU     $00B6           ;Modulus down-counter count register
MCCNT1          EQU     $00B7           ;Modulus down-counter count register
TC0HA           EQU     $00B8           ;TIC holding register 0 A
TC0HB           EQU     $00B9           ;TIC holding register 0 B
TC1HA           EQU     $00BA           ;TIC holding register 1 A
TC1HB           EQU     $00BB           ;TIC holding register 1 B
TC2HA           EQU     $00BC           ;TIC holding register 2 A
TC2HB           EQU     $00BD           ;TIC holding register 2 B
TC3HA           EQU     $00BE           ;TIC holding register 3 A
TC3HB           EQU     $00BF           ;TIC holding register 3 B
SC0BDH          EQU     $00C0           ;SCI 0 baud rate control register
                                        ;high
SC0BDL          EQU     $00C1           ;SCI 0 baud rate control register
```

```
                                        ;low
SC0CR1          EQU     $00C2           ;SCI control register 1
SC0CR2          EQU     $00C3           ;SCI control register 2
SC0SR1          EQU     $00C4           ;SCI status register 1
SC0SR2          EQU     $00C5           ;SCI status register 2
SC0DRH          EQU     $00C6           ;SCI data register high
SC0DRL          EQU     $00C7           ;SCI data register low
SP0CR1          EQU     $00D0           ;SPI control register 1
SP0CR2          EQU     $00D1           ;SPI control register 2
SP0BR           EQU     $00D2           ;SPI baud rate register
SP0SR           EQU     $00D3           ;SPI status register
SP0DR           EQU     $00D5           ;SPI data register
PORTS           EQU     $00D6           ;PORTS data register
DDRS            EQU     $00D7           ;PORTS data direction register
PURDS           EQU     $00DB           ;Port S pullup/reduced drive
                                        ;register
SLOW            EQU     $00E0           ;Slow mode divider register
EEMCR           EQU     $00F0           ;EEPROM configuration register
EEPROT          EQU     $00F1           ;EEPROM block protect register
EETST           EQU     $00F2           ;EEPROM test register
EEPROG          EQU     $00F3           ;EEPROM control register
FEELCK          EQU     $00F4           ;FLASH lock control register
FEEMCR          EQU     $00F5           ;FLASH configuration register
FEETST          EQU     $00F6           ;FLASH test register
FEECTL          EQU     $00F7           ;FLASH control register
BCR1            EQU     $00F8           ;BDLC control register 1
BSVR            EQU     $00F9           ;BDLC state vector register
BCR2            EQU     $00FA           ;BDLC control register 2
BDR             EQU     $00FB           ;BDLC data register
BARD            EQU     $00FC           ;BDLC analog roundtrip delay
                                        ;register
DLCSCR          EQU     $00FD           ;Port DLC control register
PORTDLC         EQU     $00FE           ;PORTDLC data register
DDRDLC          EQU     $00FF           ;PORTDLC data direction register

*******************************************************************************
*** msCAN12 module in register block $0100 to $017F, register equates       ***
*** not included in template (MC68HC912BC32 only)                           ***
*******************************************************************************


*******************************************************************************
```

```
*** User defined equates                                                    ***
*******************************************************************************

ZERO            EQU     $8000                   ;DAC 0V output
IDLE            EQU     $06                     ;DAC idle
L1              EQU     $00                     ;DAC load first rank of latches
L2              EQU     $07                     ;DAC load second rank of latches
TAB_START       EQU     $08FF                   ;Memory location for start of
                                                ;S-contour table

*******************************************************************************
*** Note LDAC =  bit 0 /L1 = bit 1, /CS = bit 2 of PORTP                   ***
*******************************************************************************

*******************************************************************************
*** Memory map equates                                                      ***
*******************************************************************************

RAM             EQU     $0800                   ;Location of RAM in memory map
EEPROM          EQU     $0D00                   ;Location of EEPROM in memory map
FLASH           EQU     $9000                   ;Location of FLASH in memory map
SCI_RAM         EQU     $084F                   ;Location of SCI_RAM in memory map

*******************************************************************************
*** RAM variables                                                           ***
*******************************************************************************

                ORG     RAM
LP_COUNT        DS      1                       ;Memory location of loop counter
DLY_VAL         DS      2                       ;Reserve 2 bytes for DLY_VAL
TEMP            DS      2                       ;Tempory memory location
TEMP2           DS      1                       ;Tempory memory location 2
STEP_SIZE       DS      2                       ;STEP_SIZE = 0CCC / ((N+1)/2)
K               DS      1                       ;K = ((N-3)/2) + 1
K1              DS      1                       ;K1 = ((N-1)/2)
TPT             DS      2                       ;Table pointer
F_FLG           DS      1                       ;First run flag
AD_RSLT         DS      1                       ;ADC result
AD_FLG          DS      1                       ;ADC flag
AD_TAB_RSLT     DS      2                       ;ADC table lookup result
Z_OUT           DS      2                       ;Z_OUT register
WRK_DATA        DS      7                       ;Working data
```

```
                ORG     SCI_RAM
SCI_DBUFF       DS      7                       ;SCI data buffer
SCI_FLG         DS      1                       ;SCI data ready flag
SCI_PNT         DS      1                       ;SCI data buffer pointer

*****************************************************************************************
*** SCI_DBUFF table locations are arranged as follows:-                               ***
*** 1 - Program type             (8 bit)        Data range 1,2                       ***
*** 2 - Number of data points    (8 bit)        Data range 1 - 255 odd only          ***
*** 3 - Scaling factor           (16 bit)       Data range 1 - 9                     ***
*** 4 - Maximum step size        (16 bit)       Data range                           ***
*** 5 - Z_Limit                  (16 bit)       Data range                           ***
*****************************************************************************************

*****************************************************************************************
*** Macros start here                                                                 ***
*****************************************************************************************

$MACRO MACRO1

                TFR     X,D                     ;Transfer X to D
                ADDD    #$02                    ;Add $02
                TFR     D,X                     ;Transfer D to X

$MACROEND

$MACRO  MACRO2

                LDD     #ZERO                   ;Zero DAC output
                JSR     WRT_DAC

$MACROEND

$MACRO  MACRO3

                LDAB    LP_COUNT                ;LP_COUNT = LP_COUNT - 1
                DECB
                STAB    LP_COUNT

$MACROEND
```

```
*****************************************************************************************
*** Main program starts here                                                          ***
*****************************************************************************************


                ORG     FLASH                   ;Compile code into FLASH
START           LDX     #$0BFF                  ;Initialise stack pointer to $0BFF
                TXS
                JSR     INIT                    ;Initialise processor
                MACRO2                          ;Zero DAC
                CLI                             ;Clear interrupt mask


MPL             LDAB    PORTT                   ;Check PORTT bit2 = 0 (ENABLE)
                ANDB    #$04
                BEQ     MPL                     ;If B = 0 then branch MPL
                MACRO2                          ;Zero DAC
                CLR     LP_COUNT                ;Clear LP_COUNT
                LDAB    SCI_FLG
                BEQ     MPL                     ;If SCI_FLG = 0 then branch MPL
                SEI                             ;Set interrupt mask
                CLR     SCI_FLG                 ;Clear SCI_FLG
                LDX     #SCI_DBUFF              ;Load X with address of SCI_DBUFF
                LDY     #WRK_DATA               ;Load Y with address of WRK_DATA
                LDAB    #$07                    ;Load B with $07
                STAB    LP_COUNT                ;Store B at LP_COUNT
                JSR     CPY_MEM                 ;Copy memory
                LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDAB    0,X                     ;Load B with data at 0,X
                CMPB    #$01                    ;If B = 1 then branch MPL1
                BEQ     MPL1
                BRA     MPL2                    ;Else branch MPL2
MPL1            JSR     CAL_VAR                 ;Calculate variables for linear
                JSR     CAL_LIN                 ;Calculate linear S-Contour
                JSR     SCALE                   ;Scale S-Contour table
                JSR     CORRECT                 ;Correct for DAC 0V
                BRA     MPL3                    ;Branch MPL3
MPL2            JSR     CAL_VAR1                ;Calculate variables for non linear
                JSR     CAL_NLIN                ;Calculate non linear table
                BRA     MPL3                    ;Branch MPL3
MPL3            LDAB    F_FLG                   ;If F_FLG = 1 then branch MPL4
                CMPB    #$01
```

```
                BEQ     MPL4
                CLI                             ;Clear interrupt mask
                BRA     MPL                     ;Branch MPL
MPL4            LDAB    #$03                    ;Enable TIC0 and TIC1 interrupts
                STAB    TMSK1
                CLR     F_FLG                   ;Clear F_FLG
                CLI                             ;Clear interrupt mask
                BRA     MPL                     ;Branch MPL

*******************************************************************************
*** Subroutines start here                                                  ***
*******************************************************************************

*******************************************************************************
*** Subroutine:  INIT                                                       ***
*** Author:      J R Franks                                                 ***
*** Date:        12th April 2002                                            ***
*** Function:    Initialise the processor and RAM variables                 ***
*******************************************************************************

INIT            LDX     #RAM+1                  ;Load X with RAM + 1
                LDAB    #$19                    ;LP_COUNT = $19
                STAB    LP_COUNT
                JSR     CLR_MEM                 ;Clear memory
                LDX     #SCI_RAM                ;Load X with SCI_RAM
                LDAB    #$09                    ;LP_COUNT = $09
                STAB    LP_COUNT
                JSR     CLR_MEM                 ;Clear memory
                LDX     #TAB_START              ;Load X with TAB_START
                LDAB    #$FF                    ;LP_COUNT = $FF
                STAB    LP_COUNT
                JSR     CLR_MEM                 ;Clear memory
                LDD     #TAB_START
                ADDD    #$FF
                TFR     D,X                     ;Load X with TAB_START + $FF
                LDAB    #$FF                    ;LP_COUNT = $FF
                STAB    LP_COUNT
                JSR     CLR_MEM                 ;Clear memory
                MOVB    #$FF,DDRA               ;Set PORTA output
                MOVB    #$FF,DDRB               ;Set PORTB output
                MOVB    #$07,DDRP               ;Set PORTP bits 0,1 and 2 output
                MOVB    #IDLE,PORTP             ;Set PORTP to IDLE
```

```
                LDAB    TFLG1                   ;Clear TIC0 and TIC1 pending interrupt
                ORAB    #$83
                STAB    TFLG1
                MOVB    #$80,TIOS               ;Channels 0 - 6 IC,Channel 7 OC
                MOVB    #$80,TSCR               ;Timer enabled
                MOVB    #$0A,TCTL4              ;TIC0, TIC1 falling edge sensitive
                MOVB    #$23,TMSK2              ;Pull-up enabled, prescaler clk/8
                MOVB    #$80,ATDCTL2            ;ADPU=1
                LDD     #100                    ;Wait 100uS for ADC to power up
                STD     DLY_VAL
                JSR     GP_DLY
                MOVB    #$00,ATDCTL3            ;Continue to operate ADC in background
                                                ;mode
*** SCI setup starts here ***

                MOVB    #$34,SC0BDL             ;Baud Rate 9600
                MOVB    #$00,SC0CR1             ;8bit Data format,parity disable
                MOVB    #$24,SC0CR2             ;RIE enable,Reciever enable
                LDAB    SC0SR1                  ;Clear SCI flags
                LDAB    SC0DRL

*** Ends here ***

                LDAB    #$01                    ;Set first run flag
                STAB    F_FLG
                RTS                             ;Return from subroutine

*********************************************************************************
*** Subroutine:  CPY_MEM                                                      ***
*** Author:      J R Franks                                                   ***
*** Date:        3rd July 2002                                                ***
*** Function:    Transfer N memrory locations whos source address is specified by X   ***
***              to a destination address specified by Y                      ***
*********************************************************************************

CPY_MEM         LDAB    LP_COUNT                ;Load B with LP_COUNt
                BEQ     EXIT_CPY_MEM            ;If LP_COUNT = 0 then branch EXIT_CPY_MEM
                LDAB    0,X                     ;Load B with data at 0,X
                STAB    0,Y                     ;Store B at 0,Y
                INX                             ;Increment X
                INY                             ;Increment Y
                LDAB    LP_COUNT                ;LP_COUNT = LP_COUNT - 1
```

```
                DECB
                STAB    LP_COUNT
                BRA     CPY_MEM                 ;Branch CPY_MEM
EXIT_CPY_MEM    CLR     LP_COUNT                ;Clear LP_COUNT
                RTS                             ;Return from subroutine

*******************************************************************************
*** Subroutine:  CLR_MEM                                                    ***
*** Author:      J R Franks                                                 ***
*** Date:        27th May 2002                                              ***
*** Function:    Clear N memory locations pointed to by X                   ***
*******************************************************************************

CLR_MEM         LDAB    LP_COUNT                ;Load B with LP_COUNT
                BEQ     EXIT_CLR_MEM            ;If LP_COUNT = 0 then branch EXIT_CLR_MEM
                CLR     0,X                     ;Clear memory at 0,X
                DECB                            ;LP_COUNT = LP_COUNT - 1
                STAB    LP_COUNT
                INX                             ;Increment X
                BRA     CLR_MEM                 ;Branch CLR_MEM
EXIT_CLR_MEM    CLR     LP_COUNT                ;Clear LP_COUNT
                RTS                             ;Return from subroutine

*******************************************************************************
*** Subroutine:  CAL_VAR                                                    ***
*** Author:      J R Franks                                                 ***
*** Date:        22nd April 2002                                            ***
*** Function:    Calculate STEP_SIZE and K to allow for $0000 first and      ***
***              last steps                                                 ***
*******************************************************************************

*** Calculate step size (STEP_SIZE) = 0CCC / ((N-1)/2)                      ***

CAL_VAR         LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDD     #$0000                  ;Clear D
                LDAB    1,X                     ;Load B with data at 1,X, lower half of D
                SUBD    #$01                    ;D-$01
                LDX     #$02
                IDIV                            ;D/X
                LDD     #$0CCC                  ;D/X
                IDIV
                STX     STEP_SIZE               ;Store X at STEP_SIZE
```

```
*** Calculate K = ((N-3)/2)+1 (always less than $FF)                          ***

                LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDAB    1,X                     ;Load B with data at 1,X
                SUBB    #$03                    ;B-$03
                LDX     #$02
                IDIV                            ;D/X
                TFR     X,D
                ADDD    #$01                    ;D+1
                STAB    K                       ;Store B at K
                RTS                             ;Return from subroutine

*******************************************************************************
*** Subroutine:  CAL_VAR1                                                     ***
*** Author:      J R Franks                                                   ***
*** Date:        14th May 2002                                                ***
*** Function:    Calculates K1                                                ***
*******************************************************************************

*** Calculate K1 = ((N-1)/2)                                                  ***

CAL_VAR1        LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDAB    1,X                     ;Load B with data at 1,X
                SUBB    #$01                    ;B-$01
                LDX     #$02
                IDIV                            ;D/X
                TFR     X,D
                STAB    K1                      ;Store B at K1
                RTS                             ;Return from subroutine

*******************************************************************************
*** Subroutine:  CAL_LIN                                                      ***
*** Author:      J R Franks                                                   ***
*** Date:        1st May 2002                                                 ***
*** Function:    Calculate 0.1nm linear S-Contour and store in RAM starting   ***
***              at TAB_START. First and Last steps are $0000                 ***
*******************************************************************************

CAL_LIN         LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDAB    1,X                     ;Load B with data at 1,X
                SUBB    #$02                    ;B-2
```

```
                STAB    LP_COUNT                ;Store B at LP_COUNT
                LDX     #TAB_START              ;Load X with address of TAB_START
                LDD     #$0000                  ;Clear D
                STD     TEMP                    ;Clear TEMP
                STD     0,X                     ;Store D at address specified by X
                MACRO1

CAL_LIN1        LDAB    LP_COUNT                ;Load B with LP_COUNT
                BEQ     CAL_LIN2                ;If B = 0 then branch CAL_LIN2
                CMPB    K                       ;Else compare B with K
                BCS     CAL_LIN4                ;If carry flag set branch CAL_LIN4
                BRA     CAL_LIN3                ;Else branch CAL_LIN3

CAL_LIN3        LDD     TEMP                    ;Load D with TEMP
                ADDD    STEP_SIZE               ;D+STEP_SIZE
                STD     TEMP                    ;Store D at TEMP
                STD     0,X                     ;Store D at address specified by X
                BRA     CAL_LIN5                ;Branch CAL_LIN5

CAL_LIN4        LDD     TEMP                    ;Load D with TEMP
                SUBD    STEP_SIZE               ;D-STEP_SIZE
                STD     TEMP                    ;Store D at TEMP
                STD     0,X                     ;Store D at address specified by X
                BRA     CAL_LIN5                ;Branch CAL_LIN5

CAL_LIN5        MACRO3
                MACRO1
                BRA     CAL_LIN1                ;Branch CAL_LIN1

CAL_LIN2        LDD     #$0000                  ;Clear D
                STD     0,X                     ;Store D at address specified by X
                STD     TEMP                    ;Store D at TEMP
                STAB    LP_COUNT                ;Store B at LP_COUNT
                RTS                             ;Return from subroutine

*******************************************************************************
*** Subroutine:  CAL_NLIN                                                   ***
*** Author:      J R Franks                                                 ***
*** Date:        10th May 2002                                              ***
*** Function:    Calculate and store non-linear lookup table in RAM startinf at     ***
***              TAB_START                                                  ***
*******************************************************************************
```

```
CAL_NLIN        LDX     #WRK_DATA               ;Load X with address of wrk_DATA
                LDD     4,X                     ;Load D with data at 4,X
                LDX     #$00FF                  ;Load X with $00FF
                IDIV                            ;D/X
                STX     TEMP                    ;Store X at TEMP
                LDX     #TAB_START              ;Load X with address of TAB_START
                LDD     #$0000                  ;Clear D
                STD     0,X                     ;Store D at address specified by X
                MACRO1
                LDAB    #$FF                    ;Load B with $FF
                STAB    LP_COUNT                ;Store B at LP_COUNT
CAL_NLIN1       LDAB    LP_COUNT                ;Load B with LP_COUNT
                BEQ     CAL_NLIN2               ;If B = 0 then branch CAL_NLIN2
                LDD     -2,X                    ;Else load D with data at -2,X
                ADDD    TEMP                    ;D + TEMP
                STD     0,X                     ;Store D at address specified by X
                MACRO1
                MACRO3
                BRA     CAL_NLIN1               ;Branch CAL_NLIN1
CAL_NLIN2       CLR     LP_COUNT                ;Clear LP_COUNT
                LDD     #$0000                  ;Clear D
                STD     TEMP                    ;Store D at TEMP
                RTS                             ;Return from subroutine

*********************************************************************************
*** Subroutine:  SCALE                                                        ***
*** Author:      J R Franks                                                   ***
*** Date:        22nd April 2002                                              ***
*** Function:    Scale S-Contour by integer scale factor (SF)                 ***
*********************************************************************************

SCALE           LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDAB    1,X                     ;Load B with data at 1,X
                STAB    LP_COUNT                ;Store B at LP_COUNT
                LDX     #TAB_START              ;Load X with address of TAB_START
SCALE1          LDAB    LP_COUNT                ;Load B with LP_COUNT
                BEQ     SCALE2                  ;If B = 0 then branch SCALE2
                LDY     #WRK_DATA               ;Load Y with address of WRK_DATA
                LDAB    3,Y                     ;Load B with data at 3,Y
                TFR     B,Y                     ;Transfer B to Y
                LDD     0,X                     ;Load D with data at 0,X
```

```
                EMUL                            ;Multiply D and Y
                STD     0,X                     ;Store D at address specified by X
                MACRO3
                MACRO1
                BRA     SCALE1                  ;Branch SCALE1
SCALE2          CLR     LP_COUNT                ;Clear LP_COUNT
                RTS                             ;Return form subroutine

*******************************************************************************
*** Subroutine:  CORRECT                                                    ***
*** Author:      J R Franks                                                 ***
*** Date:        22nd April 2002                                            ***
*** Function:    Correct S-Countour for DAC 0 V ($8000 - Table value)        ***
*******************************************************************************

CORRECT         LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDAB    1,X                     ;Load B with data at 1,X
                STAB    LP_COUNT                ;Store B at LP_COUNT
                LDX     #TAB_START              ;Laod X with address of TAB_START
CORRECT1        LDAB    LP_COUNT                ;Laod B with LP_COUNT
                BEQ     CORRECT2                ;Branch to CORRECT2
                LDD     #ZERO                   ;Load D with ZERO
                SUBD    0,X                     ;Subtract data at 0,X from D
                STD     0,X                     ;Store D at address specified by X
                MACRO3
                MACRO1
                BRA     CORRECT1                ;Branch CORRECT1
CORRECT2        CLR     LP_COUNT                ;Clear LP_COUNT
                RTS                             ;Return from subroutine

*******************************************************************************
*** Subroutine:  WRT_DAC                                                    ***
*** Author:      J R Franks                                                 ***
*** Date:        16th April 2002                                            ***
*** Function:    Write a word of data to the AD669                          ***
*******************************************************************************

WRT_DAC         CLR     PORTA                   ;Clear PORTA
                CLR     PORTB                   ;Clear PORTB
                STAA    PORTA                   ;Store accumulator A at PORTA
                                                ;High byte (D8 - D15)
                STAB    PORTB                   ;Store accumultaor B at PORTB
```

```
                                        ;Low byte (D0 - D7)
                MOVB    #IDLE,PORTP     ;Ensure DAC idle
                MOVB    #L1,PORTP       ;Load DAC latches rank1
                MOVB    #IDLE,PORTP     ;Idle
                MOVB    #L2,PORTP       ;Load DAC latches rank2
                MOVB    #IDLE,PORTP     ;DAC idle
                LDD     #$0000          ;Clear Z_OUT
                STD     Z_OUT
                RTS                     ;Return from subroutine

*******************************************************************************
*** Subroutine:  GP_DLY                                                     ***
*** Author:      J R Franks                                                 ***
*** Date:        12th April 2002                                            ***
*** Function:    General purpose delay using TOC7                           ***
*******************************************************************************

GP_DLY          LDAB    TFLG1           ;Load B with TFLG1
                ORAB    #$80            ;OR TFLG1 with $80
                STAB    TFLG1           ;Clear flag TOC7
                LDD     TC7H            ;Load D with TC7H
                ADDD    DLY_VAL         ;Add DLY_VAL to TC7H
                STD     TC7H            ;Store D at TC7H
GP_DLY1         LDAB    TFLG1           ;Load B with TFLG1
                ANDB    #$80            ;Remove bits 7 to 1
                BEQ     GP_DLY1         ;Branch to GP_DLY1 if flag not set
                RTS                     ;Return from subroutine

*******************************************************************************
*** Subroutine:  ADCONV                                                     ***
*** Author:      J R Franks                                                 ***
*** Date:        10th May 2002                                              ***
*** Function:    Convert analog voltage on PORTAD0 to digital equiverlent   ***
*******************************************************************************

ADCONV          CLRB                    ;Start conversion S8CM=0,SCAN=0,MULT=0,
                STAB    ATDCTL5         ;PORTAD0
ADCONV1         LDAB    ATDSTATH        ;Load B with ATDSTATH
                ANDB    #$80            ;Remove lower 7 bits
                BEQ     ADCONV1         ;If SCF = 0 then branch ADCONV1
                LDD     ADRx0H          ;Load D with ADRx0H
                STAA    AD_RSLT         ;Store A at AD_RSLT
```

```
                RTS                             ;Return from subroutine

*******************************************************************************
*** Interrupt routines start here                                           ***
*******************************************************************************

*******************************************************************************
*** Subroutine:  TIC0_INT                                                   ***
*** Author:      J R Franks                                                 ***
*** Date:        23rd April 2002                                            ***
*** Function:    Write appropriate data to DAC on interrupt                 ***
*******************************************************************************

TIC0_INT        LDAB    TFLG1                   ;Clear IC0 flag
                ORAB    #$01
                STAB    TFLG1
                LDAB    PORTT                   ;Check PORTT bit 2 = 0 (ENABLE)
                ANDB    #$04
                BNE     EXIT_TIC0               ;If bit 2 = 1 then branch EXIT_TIC0
                LDX     #WRK_DATA
                LDAB    0,X                     ;Load B with data at 0,X
                CMPB    #$01
                BEQ     TIC0_INT1               ;If P_FLG = 1 then branch TIC0_INT1
                CMPB    #$02
                BEQ     TIC0_INT4               ;If P_FLG = 2 then branch TIC0_INT4
                BRA     EXIT_TIC0               ;Else branch EXIT_TIC0

TIC0_INT1       LDAB    LP_COUNT                ;Load B with LP_COUNT
                BEQ     TIC0_INT2               ;If LP_COUNT = 0 then branch TIC0_INT2
                DECB
                STAB    LP_COUNT                ;LP-COUNT = LP_COUNT - 1
                BRA     TIC0_INT3               ;Branch TIC0_INT3
TIC0_INT2       LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDAB    1,X                     ;Load with data at 1,X
                DECB
                STAB    LP_COUNT                ;LP_COUNT = LP_COUNT - 1
                LDD     #TAB_START              ;Load D with TAB_START
                STD     TPT                     ;Store D at TPT
TIC0_INT3       LDX     TPT                     ;Load X with TPT
                LDD     0,X                     ;Load D with data at 0,X
                JSR     WRT_DAC                 ;Write to DAC
                TFR     X,D                     ;Transfer X to D
```

```
                ADDD    #$02                    ;Add $02 to D
                STD     TPT                     ;Store D at TPT
                BRA     EXIT_TIC0               ;Branch EXIT_TIC0


EXIT_TIC0       RTI                             ;Return from interrupt

TIC0_INT4       LDAB    AD_FLG                  ;Load B with AD_FLG
                CMPB    #$01
                BEQ     TIC0_INT5               ;If AD_FLG = 1 then branch TIC0_INT5
                JMP     EXIT_TIC0               ;Else jump EXIT_TIC0
TIC0_INT5       LDAB    LP_COUNT                ;Load B with LP_COUNT
                BEQ     TIC0_INT6               ;If B = 0 then branch TIC0_INT6
                DECB
                STAB    LP_COUNT                ;LP_COUNT = LP_COUNT - 1
                BRA     TIC0_INT7               ;Branch TIC0_INT7
TIC0_INT6       LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDAB    1,X                     ;Load B with data at 1,X
                DECB
                STAB    LP_COUNT                ;LP_COUNT = LP_COUNT - 1
                CLR     TEMP                    ;Clear TEMP
                CLR     TEMP+1
                CLR     TEMP2                   ;Clear TEMP2
TIC0_INT7       LDAB    LP_COUNT                ;Load B with LP_COUNT
                CMPB    #$01
                BEQ     TIC0_INT15              ;If LP_COUNT = 1 then branch TIC0_INT15
                CMPB    K1
                BLS     TIC0_INT9               ;If B <= K1 then branch TIC0_INT9
                LDD     TEMP                    ;Load D with TEMP
                ADDD    AD_TAB_RSLT             ;Add to D AD_TAB_RSLT
                STD     TEMP                    ;Store D at TEMP
                BCS     TIC0_INT8               ;If carry flag = 1 then branch TIC0_INT8
                LDAB    TEMP2                   ;Load B with TEMP2
                BEQ     TIC0_INT12              ;If TEMP2 = 0 then branch TIC0_INT12
                BRA     TIC0_INT13              ;Else branch TIC0_INT13
TIC0_INT8       LDAB    TEMP2                   ;TEMP2 = TEMP2 + 1
                INCB
                STAB    TEMP2
                BRA     TIC0_INT13              ;Branch TIC0_INT13
TIC0_INT9       LDD     TEMP                    ;Load D with TEMP
                SUBD    AD_TAB_RSLT             ;Subtract from D AD_TAB_RSLT
                STD     TEMP                    ;Store D at TEMP
```

```
                BCS     TIC0_INT10              ;If carry flag = 1 then branch TIC0_INT10
                LDAB    TEMP2                   ;Load B with TEMP2
                BEQ     TIC0_INT12              ;If TEMP2 = 0 then branch TIC0_INT12
                BRA     TIC0_INT13              ;Else branch TIC0_INT13
TIC0_INT10      LDAB    TEMP2                   ;Load B with TEMP2
                BEQ     TIC0_INT15              ;If TEMP2 = 0 then branch TIC0_INT15
                CMPB    #$01
                BEQ     TIC0_INT11              ;If TEMP2 = $01 then branch TIC0_INT11
                DECB                            ;TEMP2 = TEMP2 - 1
                STAB    TEMP2
                BRA     TIC0_INT13              ;Branch TIC0_INT13
TIC0_INT11      LDAB    TEMP2                   ;TEMP2 = TEMP2 - 1
                DECB
                STAB    TEMP2
TIC0_INT12      LDD     TEMP                    ;Load D with TEMP
                LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                CPD     6,X
                BCS     TIC0_INT14              ;If D <= Z_LIMIT then branch TIC0_INT14
TIC0_INT13      LDX     #WRK_DATA               ;Load X with address of WRK_DATA
                LDD     6,X                     ;Load D with Z_LIMIT
                STD     Z_OUT                   ;Store D at Z_OUT
                BRA     TIC0_INT16              ;Branch TIC0_INT16
TIC0_INT14      LDD     TEMP                    ;Load D with TEMP
                STD     Z_OUT                   ;Store D at Z_OUT
                BRA     TIC0_INT16              ;Branch TIC0_INT16
TIC0_INT15      LDD     #$0000                  ;Load D with $0000
                STD     Z_OUT                   ;Store D at Z_OUT
                CLR     TEMP                    ;Clear TEMP
                CLR     TEMP+1
TIC0_INT16      LDD     #ZERO                   ;Load D with ZERO
                SUBD    Z_OUT                   ;Subtract Z_OUT
                JSR     WRT_DAC                 ;Write to DAC
                CLR     AD_FLG                  ;Clear AD_FLG
                JMP     EXIT_TIC0               ;Jump EXIT_TIC0

*******************************************************************************
*** Subroutine:  TIC1_INT                                                   ***
*** Author:      J R Franks                                                 ***
*** Date:        10th May 2002                                              ***
*** Function:    Obtain data from the ADC on interrupt and look up AD_TAB_RSLT ***
*******************************************************************************
```

```
TIC1_INT        LDAB    TFLG1                   ;Clear TIC1 flag
                ORAB    #$02
                STAB    TFLG1
                LDAB    PORTT                   ;Check POTRT bit2 = 0 (ENABLE)
                ANDB    #$04
                BNE     EXIT_TIC1               ;If bit2 = 1 then branch EXIT_TIC1
                LDAB    AD_FLG                  ;If AD_FLG = 1 then branch EXIT_TIC1
                ANDB    #$01
                BNE     EXIT_TIC1
                JSR     ADCONV                  ;Get data from ADC
                LDD     #$0000                  ;Clear D
                LDAB    AD_RSLT                 ;Load D with AD_RSLT
                LDY     #$02                    ;Load Y with $02
                EMUL                            ;Multiply D and Y
                ADDD    #TAB_START              ;Add TAB_START to D
                TFR     D,X                     ;Transfer D to X
                LDD     0,X                     ;Load D from address specified by X
                STD     AD_TAB_RSLT             ;Store D at AD_TAB_RSLT
                LDAB    #$01                    ;Set AD_FLG
                STAB    AD_FLG
EXIT_TIC1       RTI                             ;Return from interrupt

*********************************************************************************
*** Subroutine: SCI_INT                                                       ***
*** Author:     J R Franks                                                    ***
*** Date:       3rd July 2002                                                 ***
*** Function:   SCI interrupt routine                                         ***
*********************************************************************************
*SCI_INT
*                LDD     #SCI_DBUFF
*                ADDD    SCI_PNT
*                TFR     D,X
*                LDAB    SC0SR1
*                LDAB    SC0DRL
*                STAB    0,X



*                INX
*                LDAB    #$07
*                STAB    SCI_PNT
*SCI_INT1        LDAB    SCI_PNT
```

```
*                BEQ    EXIT_SCI_INT
*SCI_INT2        LDAB   SC0SR1
*                ANDB   #$20
*                BEQ    SCI_INT2
*                LDAB   SC0DRL
*                STAB   0,X
*                INX
*                LDAB    SCI_PNT
*                DECB
*                STAB    SCI_PNT
*                BRA     SCI_INT1
*EXIT_SCI_INT    RTI


SCI_INT          LDAB   SC0SR1                  ;Clear interrupt flag
                 ANDB   #$20
                 BEQ    EXIT_SCI_INT

                 LDAB   SCI_PNT
                 CMPB   #$08
                 BEQ    SCI_INT1
                 BRA    SCI_INT2
SCI_INT1         LDX    #SCI_DBUFF
                 CLR    SCI_PNT
                 BRA    SCI_INT3
SCI_INT2         LDD    #SCI_DBUFF
                 ADDD   SCI_PNT
                 TFR    D,X
                 BRA    SCI_INT3
SCI_INT3         LDAB   SC0SR1
                 LDAB   SC0DRL
                 STAB   0,X
                 LDAB   SCI_PNT
                 INCB
                 STAB   SCI_PNT
EXIT_SCI_INT     RTI
*****************************************************************************************
*** Vector table                                                                      ***
*****************************************************************************************
```

```
ORG     $FFD6                   ;SCI interrupt
DW      SCI_INT

ORG     $FFEE                   ;Timer channel 0 interrupt
DW      TIC0_INT

ORG     $FFEC                   ;Timer channel 1 interrupt
DW      TIC1_INT

ORG     $FFFE                   ;Reset vector
DW      START
```

Appendix B2 – Graphical User Interface Code.

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ComCtrls, StdCtrls, SerialNG;

type
  TMainDLG = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Program1: TMenuItem;
    Communication1: TMenuItem;
    Help1: TMenuItem;
    Save1: TMenuItem;
    Open1: TMenuItem;
    Exit1: TMenuItem;
    Linear1: TMenuItem;
    NonLinear1: TMenuItem;
    SerialPortSettings1: TMenuItem;
    SendData1: TMenuItem;
    HelpIndex1: TMenuItem;
    About1: TMenuItem;
    StatusBar1: TStatusBar;
    SaveDialog: TSaveDialog;
    OpenDialog: TOpenDialog;
    SerialPortNG1: TSerialPortNG;
    N1: TMenuItem;
    N2: TMenuItem;
    procedure Linear1Click(Sender: TObject);
    procedure NonLinear1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure SendData1Click(Sender: TObject);
    procedure About1Click(Sender: TObject);
    procedure SerialPortSettings1Click(Sender: TObject);
    procedure Save1Click(Sender: TObject);
    procedure Open1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
```

```
  Program_type :Integer;
  Data_Points :Integer;
  Scaling_Factor :Integer;
  Max_Step_Size :Integer;
  Z_Limit :Integer;
  SerialNGBasicDLGOpen:Boolean;  { Public declarations }
  end;

var
  MainDLG: TMainDLG;

implementation

uses Unit2, Unit3, Unit7, Maths_Routines, SerialNGBasic;

{$R *.dfm}

procedure TMainDLG.Linear1Click(Sender: TObject);
begin
  Program_Type:=1;
  LinearDLG.Show;
  NonLinear1.Enabled:=False;
  Save1.Enabled:=True;
end;

procedure TMainDLG.NonLinear1Click(Sender: TObject);
begin
  Program_Type:=2;
  NonLinearDLG.Show;
  Linear1.Enabled:=False;
  Save1.Enabled:=True;
end;

procedure TMainDLG.Exit1Click(Sender: TObject);
begin
  if MessageDlg('Do You Wish To Exit ?', mtConfirmation, [mbYes, mbNo, mbCancel], 0) = mrYes then
    begin
      SerialPortNG1.Active:=False;
      MainDLG.Close;
    end;
end;
```

```
procedure TMainDLG.SendData1Click(Sender: TObject);

var Hex_String,Temp_String,String_Out:String;
    n,m:integer;

begin
  if MessageDlg('Send Data To MC68HC12', mtInformation, [mbYes, mbNo, mbCancel], 0) = mrYes then
    begin
      String_Out:='0';
      n:=2;
      Hex_String:=IntToHex(MainDLG.Program_type,2);//Program_Type
      HexToInt(n,Hex_string,m);
      String_Out:=String_Out+(Chr(m));

      Hex_String:=IntToHex(MainDLG.Data_Points,2);//Data_Points
      HexToInt(n,Hex_String,m);
      String_Out:=String_Out+(Chr(m));

      Hex_String:=IntToHex(MainDLG.Scaling_Factor,4);//Scaling_Factor
      Temp_String:=Hex_String[1]+ Hex_String[2];
      HexToInt(n,Temp_String,m);
      String_Out:=String_Out+(Chr(m));
      Temp_String:=Hex_String[3]+ Hex_String[4];
      HexToInt(n,Temp_String,m);
      String_Out:=String_Out+(Chr(m));

      Hex_String:=IntToHex(MainDLG.Max_Step_Size,4);//Max_Step_size
      Temp_String:=Hex_String[1]+ Hex_String[2];
      HexToInt(n,Temp_String,m);
      String_Out:=String_Out+(Chr(m));
      Temp_String:=Hex_String[3]+ Hex_String[4];
      HexToInt(n,Temp_String,m);
      String_Out:=String_Out+(Chr(m));

      Hex_String:=IntToHex(MainDLG.Z_Limit,4);//Z_Limit
      Temp_String:=Hex_String[1]+ Hex_String[2];
      HexToInt(n,Temp_String,m);
      String_Out:=String_Out+(Chr(m));
      Temp_String:=Hex_String[3]+ Hex_String[4];
      HexToInt(n,Temp_String,m);
      String_Out:=String_Out+(Chr(m));
```

```
//      String_Out:=String_Out+#$0d+#$0a;
      String_Out:='';
      String_Out:='A';
      SerialPortNG1.SendString(String_Out);
    end;
end;

procedure TMainDLG.About1Click(Sender: TObject);
begin
  AboutBox.Show;
end;

procedure TMainDLG.SerialPortSettings1Click(Sender: TObject);
begin
  SerialNGBasicDLG.SetDLGData(SerialPortNG1);
  SerialNGBasicDLG.GetDLGData(SerialPortNG1);
  SerialNGBasicDLG.ApplyBtn.Enabled:=False;
  SerialNGBasicDLG.Show;
end;

Procedure DataValidation(Input:String;Var Err_String:String;Min,Max:Integer;Var Output:Integer;Var
Err_Flg:Boolean);

var Trim_String:String;
    Trim_Char:Char;
    String_Length,m:Integer;

begin
  Err_Flg:=False;
  Err_String:='';
  Trim_String:='';
  Output:=0;
  if not (Input = '') then
  begin
    String_Length:=ByteToCharLen(Input,10);
    For m:=1 to String_Length do
    begin
      Trim_Char:=Input[m];
      if not ((Trim_Char >='0') and (Trim_Char <='9')) then
      begin
        Err_Flg:=True;
        Err_String:='File Open Error: Non Numeric Data Value ';
```

```
      end
      else
      begin
        Trim_String:=Trim_String+Trim_Char;
      end;
    end;
    if not (Err_Flg = True) then
    begin
      m:=StrToInt(Trim_String);
      if ((m < Min) or (m > Max)) then
      begin
        Err_Flg:=True;
        Err_String:='File Open Error: Data Value Invalid ';
      end
      else
      begin
        Output:=m;
      end;
    end;
  end
  else
  begin
    Err_Flg:= True;
    Err_String:='File Open Error: Empty Data Field ';
  end;
end;

procedure TMainDLG.Save1Click(Sender: TObject);

var  SaveFile:textfile;
     SaveString:String;

begin
  SaveDialog := TSaveDialog.Create(Self);
  SaveDialog.Filter:='Data Files (*.Dat)|*.Dat';
  SaveDialog.FileName:=OpenDialog.FileName;
  if SaveDialog.Execute then
  begin
    AssignFile(SaveFile,SaveDialog.FileName+SaveDialog.DefaultExt);// Current File name
    rewrite(SaveFile);
    SaveString:='FileName: '+SaveDialog.FileName+#$0d#$0a;
    writeln(SaveFile,SaveString);
```

```
  SaveString:= 'Date/Time: '+ DateTimeToStr(Now)+#$0d#$0a;// Current Date and Time
  Writeln(SaveFile,SaveString);
  SaveString:='Data Section'+#$0d#$0a;
  Writeln(SaveFile,SaveString);
  SaveString:='Program Type:'+#$0d#$0a+IntToStr(Program_Type);
  Writeln(SaveFile,SaveString);
  SaveString:='Data Points:'+#$0d#$0a+IntToStr(Data_Points);
  Writeln(SaveFile,SaveString);
  SaveString:='Scaling Factor:'+#$0d#$0a+IntToStr(LinearDLG.CBScalingFactor.ItemIndex);
  Writeln(SaveFile,SaveString);
  SaveString:='Maximum Step Size:'+#$0d#$0a+IntToStr(NonLinearDLG.CBStepsize.ItemIndex);
  Writeln(SaveFile,SaveString);
  SaveString:='Z_Limit:'+#$0d#$0a+IntToStr(NonLinearDLG.CBZlimit.ItemIndex)+#$0d#$0a;
  Writeln(SaveFile,SaveString);
  SaveString:='Enviromental Parameters'+#$0d#$0a;
  Writeln(SaveFile,SaveString);
  SaveString:='Serial Port DLG Open(1)/Close(0):';
  Writeln(SaveFile,SaveString);
  if SerialNGBasicDLGOpen = True then
  begin
    SaveString:='1';
    Writeln(SaveFile,SaveString);
  end
  else
  begin
    SaveString:='0';
    Writeln(SaveFile,SaveString);
  end;
  SaveString:='Comm Port:'+#$0d#$0a+IntToStr(SerialNGBasicDLG.CBPort.ItemIndex);
  Writeln(SaveFile,SaveString);
  SaveString:='Baud Rate:'+#$0d#$0a+IntToStr(SerialNGBasicDLG.CBBaud.ItemIndex);
  Writeln(SaveFile,SaveString);
  SaveString:='Parity:'+#$0d#$0a+IntToStr(SerialNGBasicDLG.CBParity.ItemIndex);
  Writeln(SaveFile,SaveString);
  SaveString:='Flow Control:'+#$0d#$0a+IntToStr(SerialNGBasicDLG.CBFlow.ItemIndex);
  Writeln(SaveFile,SaveString);
  SaveString:='Data Bits:'+#$0d#$0a+IntToStr(SerialNGBasicDLG.CBData.ItemIndex);
  Writeln(SaveFile,SaveString);
  SaveString:='Stop Bits:'+#$0d#$0a+IntToStr(SerialNGBasicDLG.CBStop.ItemIndex);
  Writeln(SaveFile,SaveString);
  CloseFile(SaveFile);
end;
```

```
  SaveDialog.Free;
end;

procedure TMainDLG.Open1Click(Sender: TObject);

var Open_File:textfile;
    Open_String,Error_String:String;
    Error_Flag:Boolean;
    Open_Data,n,m,Scale_Factor_Index,Max_Step_Size_Index,Z_Limit_Index,Serial_Port_DLG_Open:Integer;

Comm_Port_Index,Baud_Rate_Index,Parity_Index,Flow_Control_Index,Data_Bits_Index,Stop_Bits_Index:Integer;

begin
  if SerialNGBasicDLGOpen = True then
  begin
    SerialNGBasicDLG.Close;
  end;
  if Program_Type = 1 then
  begin
    LinearDLG.Close;
    NonLinear1.Enabled:=True;
    Save1.Enabled:=False;
  end
  else if Program_Type = 2 then
  begin
    NonLinearDLG.Close;
    Linear1.Enabled:=True;
    Save1.Enabled:=False;
  end;
  OpenDialog := TOpenDialog.Create(Self);
  OpenDialog.Filter:='Data Files (*.Dat)|*.Dat';
  if OpenDialog.Execute then
  begin
    n:=0;
    Error_Flag:=False;
    AssignFile(Open_File,OpenDialog.FileName+OpenDialog.DefaultExt);
    Reset(Open_File);
    while ((not eof(Open_File)) and (not Error_Flag)) do
      begin
        Readln(Open_File,Open_String);
        case n of
          7 :begin
```

```
      DataValidation(Open_String,Error_String,1,2,Open_data,Error_Flag);
      Error_String:=Error_String+'"Program Type"';
    end;
 9 :begin
      DataValidation(Open_String,Error_String,1,255,Open_data,Error_Flag);
      Error_String:=Error_String+'"Data Points"';
      m:=Open_Data mod 2;
      if m = 0 then
      begin
        Error_Flag:=True;
        Error_String:='File Open Error: Data Value Invalid "Data Points"';
      end;
    end;
11:begin
      DataValidation(Open_String,Error_String,0,8,Open_data,Error_Flag);
      Error_String:=Error_String+'"Scaling Factor"';
    end;
13:begin
      DataValidation(Open_String,Error_String,0,3,Open_data,Error_Flag);
      Error_String:=Error_String+'"Max Step Size"';
    end;
15:begin
      DataValidation(Open_String,Error_String,0,19,Open_data,Error_Flag);
      Error_String:=Error_String+'"Z Limit"';
    end;
 20:begin
       DataValidation(Open_String,Error_String,0,1,Open_data,Error_Flag);
       Error_String:=Error_String+'"Serial Port DLG"';
     end;
 22:begin
       DataValidation(Open_String,Error_String,0,7,Open_data,Error_Flag);
       Error_String:=Error_String+'"Comm Port"';
     end;
 24:begin
       DataValidation(Open_String,Error_String,0,14,Open_data,Error_Flag);
       Error_String:=Error_String+'"Baud Rate"';
     end;
 26:begin
       DataValidation(Open_String,Error_String,0,4,Open_data,Error_Flag);
       Error_String:=Error_String+'"Parity"';

     end;
```

```
        28:begin
             DataValidation(Open_String,Error_String,0,3,Open_data,Error_Flag);
             Error_String:=Error_String+'"Flow Control"';
           end;
        30:begin
             DataValidation(Open_String,Error_String,0,4,Open_data,Error_Flag);
             Error_String:=Error_String+'"Data Bits"';
           end;
        32:begin
             DataValidation(Open_String,Error_String,0,2,Open_data,Error_Flag);
             Error_String:=Error_String+'"Stop Bits"';
           end;
      end;
      n:=n+1;
    end;
  Reset(Open_File);
  if Error_Flag = True Then
    StatusBar1.Panels[1].Text:=Error_String
  else
    begin
      n:=0;
      While (not eof(Open_File)) do
        begin
          Readln(Open_File,Open_String);
          Case n of
            7 : Program_Type:=StrToInt(Open_String);
            9 : Data_Points:=StrToInt(Open_String);
            11: Scale_Factor_Index:=StrToInt(Open_String);
            13: Max_Step_Size_Index:=StrToInt(Open_String);
            15: Z_Limit_Index:=StrToInt(Open_String);
            20: Serial_Port_DLG_Open:=StrToInt(Open_String);
            22: Comm_Port_Index:=StrToInt(Open_String);
            24: Baud_Rate_Index:=StrToInt(Open_String);
            26: Parity_Index:=StrToInt(Open_String);
            28: Flow_Control_Index:=StrToInt(Open_String);
            30: Data_Bits_Index:=StrToInt(Open_String);
            32: Stop_Bits_Index:=StrToInt(Open_String);
          end;
        n:=n+1;
        end;
    end;
    if not (Error_Flag = True) Then
```

```
      begin
        StatusBar1.Panels[1].Text:='';
        if Program_Type = 1 then
        begin
          LinearDLG.DP_Edit1.Text:=IntToStr(Data_Points);
          LinearDLG.CBScalingFactor.ItemIndex:=Scale_Factor_Index;
          Linear1.Click;
        end
        else if ((Program_Type = 2) and (Error_Flag=False)) then
        begin
          NonLinearDLG.DP_Edit2.Text:=IntToStr(Data_Points);
          NonLinearDLG.CBStepsize.ItemIndex:=Max_Step_Size_Index;
          NonLinearDLG.CBZlimit.ItemIndex:=Z_Limit_Index;
          NonLinear1.Click;
        end;
        SerialNGBasicDLG.CBPort.ItemIndex:=Comm_Port_Index;
        SerialNGBasicDLG.CBBaud.ItemIndex:=Baud_Rate_Index;
        SerialNGBasicDLG.CBData.ItemIndex:=Data_Bits_Index;
        SerialNGBasicDLG.CBStop.ItemIndex:=Stop_Bits_Index;
        SerialNGBasicDLG.CBParity.ItemIndex:=Parity_Index;
        SerialNGBasicDLG.CBFlow.ItemIndex:=Flow_Control_Index;
        SerialNGBasicDLG.GetDLGData(SerialPortNG1);
        if Serial_Port_DLG_Open = 1 then
        begin
          SerialPortSettings1.Click;
        end;
      end;
      CloseFile(Open_File);
  end;
  OpenDialog.Free;
end;

procedure TMainDLG.FormCreate(Sender: TObject);
begin
  SendData1.Enabled:=False;
  Save1.Enabled:=False;
  SerialPortNG1.Active:=True;
end;

end.
```

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Spin, ExtCtrls, Menus;

type
  TLinearDLG = class(TForm)
    CloseBtn: TButton;
    DP_Edit1: TEdit;
    ResetBtn: TButton;
    CBScalingFactor: TComboBox;
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    procedure CloseBtnClick(Sender: TObject);
    procedure CBScalingFactorChange(Sender: TObject);
    procedure ResetBtnClick(Sender: TObject);
    procedure DP_Edit1KeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormCreate(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure DP_Edit1Exit(Sender: TObject);
    procedure DP_Edit1Enter(Sender: TObject);
    procedure DP_Edit1Change(Sender: TObject);
    procedure FormDeactivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  LinearDLG: TLinearDLG;

implementation

uses Unit1;

{$R *.dfm}
```

```
procedure TLinearDLG.CloseBtnClick(Sender: TObject);
begin
  LinearDLG.Close;
  MainDLG.Save1.Enabled:=False;
  MainDLG.SendData1.Enabled:=False;
  MainDLG.NonLinear1.Enabled:=True;
  MainDLG.StatusBar1.Panels[1].Text :='';
end;

procedure TLinearDLG.CBScalingFactorChange(Sender: TObject);
const a = $0CCC;
var i,k:integer;
begin
  i:=LinearDLG.CBScalingFactor.ItemIndex + 1;
  k:=i*a;
  MainDLG.Scaling_Factor:=k;
end;

procedure TLinearDLG.ResetBtnClick(Sender: TObject);
begin
  DP_Edit1.Clear;
  DP_Edit1.Text:=IntToStr(1);//1 data point
  DP_Edit1.Font.Color:=ClBlack;
  MainDLG.Data_Points:=1;
  LinearDLG.CBScalingFactor.ItemIndex:=0; //1nm
  MainDLG.Scaling_Factor:=$0CCC;
  MainDLG.StatusBar1.Panels[1].Text:='';
end;

procedure TLinearDLG.DP_Edit1KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);

const DP_Min=1;
      DP_Max=255;
Var   Code, Temp, Data_Points :Integer;

begin
  Data_Points:=DP_Min;
  Temp:=0;
  if key = 13 then
  begin
```

```
    DP_Edit1.Font.Color := ClBlue;
    Val(DP_Edit1.Text,Data_Points,Code);
    if (code=0) then
    begin
      Data_Points := StrToInt(DP_Edit1.Text);
      if Data_Points < DP_Min then
      begin
        Data_Points:=DP_Min;
        DP_Edit1.Font.Color:=ClRed;
        DP_Edit1.Text:= IntToStr(DP_Min);
        MainDLG.StatusBar1.Panels[1].Text:= 'Data Entry Error: Data Point Value Must Be >=1';
      end
      else if Data_Points > DP_Max then
      begin
        Data_Points:=DP_Min;
        DP_Edit1.Font.Color:=ClRed;
        DP_Edit1.Text:= IntToStr(DP_Min);
        MainDLG.StatusBar1.Panels[1].Text:='Data Entry Error: Data Point Value Must Be <=255';
      end;
      Temp:=Data_Points MOD 2;
      if Temp = 0 then
      begin
        Data_Points:=DP_Min;
        DP_Edit1.Font.Color:=ClRed;
        DP_Edit1.Text:= IntToStr(DP_Min);
        MainDLG.StatusBar1.Panels[1].Text:='Data Entry Error: Data Point Value Must Be Odd';
      end;
    end
    else
    begin
      Data_Points:=DP_Min;
      DP_Edit1.Font.Color:=ClRed;
      DP_Edit1.Text:= IntToStr(DP_Min);
      MainDLG.StatusBar1.Panels[1].Text:='Data Entry Error: Data Point Value Must Be An Integer';
    end;
    MainDLG.Data_Points:= Data_Points;
  end;
end;

procedure TLinearDLG.FormCreate(Sender: TObject);
begin
  DP_Edit1.Text:=IntToStr(1);//1 data point
```

```
  MainDLG.Data_Points:=1;
  LinearDLG.CBScalingFactor.ItemIndex:=0; //1nm
  MainDLG.Scaling_Factor:=$0CCC;
end;

procedure TLinearDLG.FormActivate(Sender: TObject);
begin
  MainDLG.Save1.Enabled:=True;
  MainDLG.SendData1.Enabled:=True;
end;


procedure TLinearDLG.DP_Edit1Exit(Sender: TObject);
begin
  DP_Edit1.Clear;
  DP_Edit1.Text:=IntToStr(MainDLG.Data_Points);
  DP_Edit1.Font.Color:=ClBlack;
end;

procedure TLinearDLG.DP_Edit1Enter(Sender: TObject);
begin
  DP_Edit1.Font.Color:=ClRed;
end;

procedure TLinearDLG.DP_Edit1Change(Sender: TObject);
begin
  DP_Edit1.Font.color:=ClRed;
end;

procedure TLinearDLG.FormDeactivate(Sender: TObject);
begin
  DP_Edit1.Text:=IntToStr(MainDLG.Data_Points);
  DP_Edit1.Font.Color:=ClBlack;
end;

end.
```

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Spin, ExtCtrls;

type
  TNonLinearDLG = class(TForm)
    CloseBtn: TButton;
    DP_Edit2: TEdit;
    ResetBtn: TButton;
    Panel1: TPanel;
    CBZlimit: TComboBox;
    Label1: TLabel;
    Label2: TLabel;
    CBStepsize: TComboBox;
    Label3: TLabel;
    procedure CloseBtnClick(Sender: TObject);
    procedure DP_Edit2KeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure ResetBtnClick(Sender: TObject);
    procedure CBZlimitChange(Sender: TObject);
    procedure CBStepsizeChange(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure DP_Edit2Exit(Sender: TObject);
    procedure DP_Edit2Enter(Sender: TObject);
    procedure DP_Edit2Change(Sender: TObject);
    procedure FormDeactivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  NonLinearDLG: TNonLinearDLG;

implementation
```

```
uses Unit1;

{$R *.dfm}

procedure TNonLinearDLG.CloseBtnClick(Sender: TObject);
begin
  NonLinearDLG.Close;
  MainDLG.Save1.Enabled:=False;
  MainDLG.SendData1.Enabled:=False;
  MainDLG.Linear1.Enabled:=True;
  MainDLG.StatusBar1.Panels[1].Text:='';
end;


procedure TNonLinearDLG.DP_Edit2KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);

const DP_Min=1;
      DP_Max=255;
Var   Code, Temp, Data_Points :Integer;

begin
  Data_Points:=DP_Min;
  Temp:=0;
  if key = 13 then
  begin
    DP_Edit2.Font.Color:=ClBlue;
    Val(DP_Edit2.Text,Data_Points,Code);
    if (code=0) then
    begin
      Data_Points := StrToInt(DP_Edit2.Text);
      if Data_Points < DP_Min then
      begin
        Data_Points:=DP_Min;
        DP_Edit2.Font.Color:=ClRed;
        DP_Edit2.Text:= IntToStr(DP_Min);
        MainDLG.StatusBar1.Panels[1].Text:= 'Data Entry Error: Data Point Value Must Be >=1';
      end
      else if Data_Points > DP_Max then
      begin
        Data_Points:=DP_Min;
        DP_Edit2.Font.Color:=ClRed;
```

```
        DP_Edit2.Text:= IntToStr(DP_Min);
        MainDLG.StatusBar1.Panels[1].Text:='Data Entry Error: Data Point Value Must Be <=255';
      end;
      Temp:=Data_Points MOD 2;
      if Temp = 0 then
      begin
        Data_Points:=DP_Min;
        DP_Edit2.Font.Color:=ClRed;
        DP_Edit2.Text:= IntToStr(DP_Min);
        MainDLG.StatusBar1.Panels[1].Text:='Data Entry Error: Data Point Value Must Be Odd';
      end;
    end
    else
    begin
      Data_Points:=DP_Min;
      DP_Edit2.Font.Color:=ClRed;
      DP_Edit2.Text:= IntToStr(DP_Min);
      MainDLG.StatusBar1.Panels[1].Text:='Data Entry Error: Data Point Value Must Be An Integer';
    end;
    MainDLG.Data_Points:= Data_Points;
  end;
end;
procedure TNonLinearDLG.ResetBtnClick(Sender: TObject);
begin
  DP_Edit2.Clear;
  DP_Edit2.Text:= IntToStr(1);//1 data point
  DP_Edit2.Font.Color:=ClBlack;
  MainDLG.Data_Points:=1;
  NonLinearDLG.CBZlimit.ItemIndex:=0;//5 nm
  MainDLG.Z_Limit:=$0666;
  NonLinearDLG.CBStepsize.ItemIndex:=0;//5 nm
  MainDLG.Max_Step_Size:=$0666;
  MainDLG.StatusBar1.Panels[1].Text:='';
end;

procedure TNonLinearDLG.CBZlimitChange(Sender: TObject);
const a = $0666;
var i,k:integer;
begin
  i:=NonLinearDLG.CBZlimit.ItemIndex + 1;
  k:=i*a;
  MainDLG.Z_Limit:=k;
```

```
end;

procedure TNonLinearDLG.CBStepsizeChange(Sender: TObject);
const a = $0666;
var i,k:integer;
begin
  i:=NonLinearDLG.CBStepsize.ItemIndex + 1;
  k:=i*a;
  MainDLG.Max_Step_Size:=k;
end;

procedure TNonLinearDLG.FormCreate(Sender: TObject);
begin
  DP_Edit2.Text:=IntToStr(1);
  MainDLG.Data_Points:=1;
  NonLinearDLG.CBZlimit.ItemIndex:=0;
  MainDLG.Z_Limit:=$0666;
  NonLinearDLG.CBStepsize.ItemIndex:=0;
  MainDLG.Max_Step_Size:=$0666;
end;

procedure TNonLinearDLG.FormActivate(Sender: TObject);
begin
  MainDLG.Save1.Enabled:=True;
  MainDLG.SendData1.Enabled:=True;
end;

procedure TNonLinearDLG.DP_Edit2Exit(Sender: TObject);
begin
  DP_Edit2.Clear;
  DP_Edit2.Text:=IntToStr(MainDLG.Data_Points);
  DP_Edit2.Font.Color:=ClBlack;
end;

procedure TNonLinearDLG.DP_Edit2Enter(Sender: TObject);
begin
  DP_Edit2.Font.Color:=ClRed;
end;

procedure TNonLinearDLG.DP_Edit2Change(Sender: TObject);
begin
  DP_Edit2.Font.Color:=ClRed;
```

```
end;

procedure TNonLinearDLG.FormDeactivate(Sender: TObject);
begin
  DP_Edit2.Text:=IntToStr(MainDLG.Data_Points);
  DP_Edit2.Font.Color:=ClBlack;
end;

end.
```

```
unit Unit4;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, SerialNG, StdCtrls, ExtCtrls;

type
  TSerialPortParametersDLG = class(TForm)
    SerialPortNG1: TSerialPortNG;
    CloseBtn: TButton;
    ApplyBtn: TButton;
    Panel1: TPanel;
    CBPort: TComboBox;
    CBBaud: TComboBox;
    CBParity: TComboBox;
    CBFlow: TComboBox;
    CBDatabits: TComboBox;
    CBStopbits: TComboBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    DefaultBtn: TButton;
    TestBtn: TButton;
    procedure CloseBtnClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure ApplyBtnClick(Sender: TObject);
    procedure CBPortChange(Sender: TObject);
    procedure CBBaudChange(Sender: TObject);
    procedure CBParityChange(Sender: TObject);
    procedure CBFlowChange(Sender: TObject);
    procedure CBDatabitsChange(Sender: TObject);
    procedure CBStopbitsChange(Sender: TObject);
    procedure DefaultBtnClick(Sender: TObject);
    procedure TestBtnClick(Sender: TObject);
  private
    { Private declarations }
  public
```

```
    { Public declarations }
  end;

var
  SerialPortParametersDLG: TSerialPortParametersDLG;

implementation

{$R *.dfm}



procedure TSerialPortParametersDLG.CloseBtnClick(Sender: TObject);
begin
  SerialPortParametersDLG.Close;
end;

Procedure SetDLGData(SerialPortNG1 : TSerialPortNG);
var i:Integer;
begin
  i:= SerialPortParametersDLG.CBPort.Items.IndexOf(SerialPortNG1.CommPort);
  if i >= 0 then
    SerialPortParametersDLG.CBPort.ItemIndex := i
  else
    SerialPortParametersDLG.CBPort.ItemIndex := 1; //Comm2
  i:= SerialPortParametersDLG.CBBaud.Items.IndexOf(IntToStr(SerialPortNG1.BaudRate));
  if i >= 0 then
    SerialPortParametersDLG.CBBaud.ItemIndex := i
  else
    SerialPortParametersDLG.CBBaud.ItemIndex := 6; //9600
  i:= SerialPortParametersDLG.CBDatabits.Items.IndexOf(IntToStr(SerialPortNG1.DataBits)+ 'Bit');
  if i >= 0 then
    SerialPortParametersDLG.CBDatabits.ItemIndex :=i
  else
    SerialPortParametersDLG.CBDatabits.ItemIndex := 4; //8 Bit
  SerialPortParametersDLG.CBStopbits.ItemIndex := SerialPortNG1.StopBits;
  SerialPortParametersDLG.CBParity.ItemIndex := SerialPortNG1.ParityType;
  Case SerialPortNG1.FlowControl of
    fcNone :SerialPortParametersDLG.CBFlow.ItemIndex:=0;
    fcXON_XOFF :SerialPortParametersDLG.CBFlow.ItemIndex :=1;
    fcRTS_CTS :SerialPortParametersDLG.CBFlow.ItemIndex :=2;
    fcDSR_DTR :SerialPortParametersDLG.CBFlow.ItemIndex :=3;
```

```
  else
    SerialPortParametersDLG.CBFlow.ItemIndex := 0;
  end;
end;

Procedure DefaultDLGSettings(SerialPortNG1 : TSerialPortNG);
begin
  SerialPortParametersDLG.CBPort.ItemIndex := 1; //Comm1
  SerialPortParametersDLG.CBBaud.ItemIndex := 0; //9600 Baud
  SerialPortParametersDLG.CBDatabits.ItemIndex := 4; //8 Bit
  SerialPortParametersDLG.CBStopbits.ItemIndex := 0; //1 bit
  SerialPortParametersDLG.CBParity.ItemIndex := 0; //None(*)
  SerialPortParametersDLG.CBFlow.ItemIndex := 0; //None(*)
end;

Procedure GetDLGData(SerialPortNG1 : TSerialPortNG);
begin
  SerialPortParametersDLG.SerialPortNG1.CommPort :=
SerialPortParametersDLG.CBPort.Items[SerialPortParametersDLG.CBPort.ItemIndex];
  SerialPortParametersDLG.SerialPortNG1.BaudRate :=
StrToIntDef(SerialPortParametersDLG.CBBaud.Items[SerialPortparametersDLG.CBBaud.ItemIndex],9600);
  SerialPortParametersDLG.SerialPortNG1.DataBits :=
StrToIntDef(Copy(SerialPortParametersDLG.CBDatabits.Items[SerialPortParametersDLG.CBDatabits.ItemIndex],1,1
),8);
  SerialPortParametersDLG.SerialPortNG1.StopBits := SerialPortParametersDLG.CBStopbits.ItemIndex;
  SerialPortParametersDLG.SerialPortNG1.ParityType := SerialPortParametersDLG.CBParity.ItemIndex;
  SerialPortParametersDLG.SerialPortNG1.FlowControl :=
BasicFlowModes[SerialPortParametersDLG.CBFlow.ItemIndex];
  SerialPortNG1.Active :=True;
  SerialPortParametersDLG.ApplyBtn.Enabled := False;
end;


procedure TSerialPortParametersDLG.FormCreate(Sender: TObject);
begin
  SetDLGData(SerialPortNG1);
  GetDLGData(SerialPortNG1);
end;

procedure TSerialPortParametersDLG.ApplyBtnClick(Sender: TObject);
begin
  GetDLGData(SerialPortNG1);
```

```
end;


procedure TSerialPortParametersDLG.CBPortChange(Sender: TObject);
begin
  ApplyBtn.Enabled := True;
end;

procedure TSerialPortParametersDLG.CBBaudChange(Sender: TObject);
begin
  ApplyBtn.Enabled := True;
end;

procedure TSerialPortParametersDLG.CBParityChange(Sender: TObject);
begin
  ApplyBtn.Enabled := True;
end;

procedure TSerialPortParametersDLG.CBFlowChange(Sender: TObject);
begin
  ApplyBtn.Enabled := True;
end;

procedure TSerialPortParametersDLG.CBDatabitsChange(Sender: TObject);
begin
  ApplyBtn.Enabled := True;
end;

procedure TSerialPortParametersDLG.CBStopbitsChange(Sender: TObject);
begin
  ApplyBtn.Enabled := True;
end;

procedure TSerialPortParametersDLG.DefaultBtnClick(Sender: TObject);
begin
  DefaultDLGSettings(SerialPortNG1);
  ApplyBtn.Enabled := True;
end;

procedure TSerialPortParametersDLG.TestBtnClick(Sender: TObject);
begin
  SerialPortNG1.SendString('Hello');
```

```
end;

end.
```

```
unit Unit5;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TForm5 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation

{$R *.dfm}

end.
```

```
unit Unit6;

interface

uses
  SysUtils, Classes;

type
  TDataModule6 = class(TDataModule)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  DataModule6: TDataModule6;

implementation

{$R *.dfm}

end.
```

```
unit Unit7;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,
  Buttons, ExtCtrls;

type
  TAboutBox = class(TForm)
    Panel1: TPanel;
    ProgramIcon: TImage;
    OKButton: TButton;
    Panel2: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    procedure OKButtonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  AboutBox: TAboutBox;

implementation

{$R *.dfm}

procedure TAboutBox.OKButtonClick(Sender: TObject);
begin
  AboutBox.Close;
end;

end.
```

```
unit Maths_Routines;

interface

Procedure Power(var x,y,exponent:integer);
{Performs y=x^exponent}

Procedure HexToInt(var n:Integer; Hex_value:String; var Int_value:Integer);
{Performs hexadecimal to integer conversion}

Procedure HexToIntDigit(var Hex_value:Char; var Int_value:Integer);
{Performs hexadecimal to integer conversion for 1 digit}

implementation

Procedure HexToIntDigit(var Hex_value:Char; var Int_value:Integer);
begin
  Case Hex_value of
    '0' : Int_value:=0;
    '1' : Int_value:=1;
    '2' : Int_value:=2;
    '3' : Int_value:=3;
    '4' : Int_value:=4;
    '5' : Int_value:=5;
    '6' : Int_value:=6;
    '7' : Int_value:=7;
    '8' : Int_value:=8;
    '9' : Int_value:=9;
    'A' : Int_value:=10;
    'B' : Int_value:=11;
    'C' : Int_value:=12;
    'D' : Int_value:=13;
    'E' : Int_value:=14;
    'F' : Int_value:=15;
  end;
end;

Procedure Power(Var x,y,exponent:integer);
var m:integer;
begin
  y:=1;
  For m:=exponent downto 1 do
```

```
  begin
    y:=y*x;
  end;
end;

Procedure HexToInt(var n:Integer; Hex_value:String; var Int_value:Integer);
var m,x,y,z,exponent:Integer;
begin
  m:=n;
  Int_value:=0;
  For m:=n downto 1 do
  begin
    HexToIntDigit(Hex_value[m],x);
    exponent:=(n-m);
    z:=16;
    Power(z,y,exponent);
    Int_value:=Int_value+(x*y);
  end;
end;

end.
```

Appendix B3 – Macro Code.

```
;Title:          Demonstration Macro
;File name:      Demo.MAC
;Author:         J R Franks
;Date:           25th September 2003
;Function:       Performs hardware and software based variable tip-sample separaion and
;                fixed tip-sample sample separation STS.



;Data points:                    121
;Acquisition time:               1280uS (128 Samples)
;Gap voltage:                    -3 to +3 V
;Step size:                      100mV
;Status port signal 12:          Enable          (MC68HC12 Port t bit 2)
;Status port signal 13:          Step            (MC68HC12 Port t bit 0 TIC0_INT)
;Status port signal 14:          Sample          (MC68HC12 Port t bit 1 TIC1_INT)
;Status port signal 15:          Enable modulating signal from lockin
;Note: All Status port signals are active LOW


acq_time, 1280                          ;Define acquistion time 1280us
delay, 200                              ;Delay 200uS prior to loop off
feedback, 0                             ;Disable feedback loop
set_v, -3.0                             ;Set Ugap to -3.0V
Set_z, 0                                ;Set Z to regulated height
delay,200                               ;Delay 200us to allow tip sample system to stabalise

;Main code goes here

;Software based variable tip-sample separation with software based conductivity measurements
;Ramp in

set_v, -3.0
loop_times, 60                          ;Repeat next commands 60 times
        measure                         ;Measure tunnel current
        delay, 50                       ;Delay 50uS
        add_v, -0.02                    ;Add to Ugap -10mv
        measure                         ;Measure tunnel current
        add_v, 0.040                    ;Add to Ugap 20mV
        delay, 50                       ;Delay 50uS
        measure                         ;Measure tunnel current
        add_v, 0.030                    ;Add to Ugap 40mV
        add_z, -0.01
loop_end
        measure                         ;Measure tunnel current
        delay, 50                       ;Delay 50uS
```

```
        add_v, -0.02                    ;Add to Ugap -10mv
        measure                         ;Measure tunnel current
        add_v, 0.040                    ;Add to Ugap 20mV
        delay, 50                       ;Delay 50uS
        measure                         ;Measure tunnel current
        add_v, 0.030                    ;Add to Ugap 40mV
;Ramp out
loop_times, 60                          ;Repeat next commands 60 times
        measure                         ;Measure tunnel current
        delay, 50                       ;Delay 50uS
        add_v, -0.02                    ;Add to Ugap -10mv
        measure                         ;Measure tunnel current
        add_v, 0.040                    ;Add to Ugap 20mV
        delay, 50                       ;Delay 50uS
        measure                         ;Measure tunnel current
        add_v, 0.030                    ;Add to Ugap 40mV
        add_z, 0.01                     ;Add to Z
loop_end
set_v, -3.0                             ;Set Ugap to -3.0V

;Hardware based variable tip-sample separation.

set_signal, 12                          ;Enable MC68HC12 hardware
loop_times, 121                         ;Repeat next commands 121 times
        set_signal, 13                  ;Single hardware step
        delay, 50                       ;Delay 50uS
        reset_signal, 13                ;Reset signal 13
        delay, 200                      ;Delay 200uS
        measure                         ;Measure tunnel current
        delay, 50                       ;Delay 50uS
        set_signal, 15                  ;Turn on modulation
        delay, 40000                    ;Delay 40000us
        measure                         ;Measrue conductivity (EXT1)
        delay, 50                       ;Dealy 50uS
        reset_signal, 15                ;Turn off modulation
        delay, 50                       ;Delay 50uS
        add_v, 0.05                     ;Add to Ugap 50mV
loop_end
reset_signal, 12                        ;Disable hardware tip movement
set_v, -3.0                             ;Set Ugap to -3.0V

;Fixed tip-sample sample separation

set_v, -3.0                             ;Set Ugap -3.0V
loop_times, 121                         ;Repeat next commands 121 times
        delay, 50                       ;Delay 50uS
```

```
        measure                 ;Measure tunnel current
        add_v, 0.05             ;Add to Ugap 45mV
loop_end

;Reset code

reset_z                         ;Reset Z to regulated height
reset_v                         ;Reset Ugap to scan value
delay, 100                      ;Delay 100uS proir to loop on
feedback, 1                     ;Enable feedback loop
delay, 50                       ;Delay 50uS
```